

APPROVAL SHEET

Title of Dissertation: Hybrid Approaches to Color Image Quantization

Name of Candidate: Paula Julie Reitan
Doctor of Philosophy, 1999

Dissertation and Abstract Approved: _____

Dr. Charles K. Nicholas

Associate Professor

Department of Computer Science and Electrical Engineering

Date Approved: January 25, 1999

CURRICULUM VITAE

Name: Paula Julie Reitan.

Permanent Address: 5902 Grenfell Loop, Bowie, Maryland 20720.

Degree and date to be conferred: Doctor of Philosophy, 1999.

Date of Birth: October 6, 1963.

Place of Birth: Omaha, Nebraska.

Secondary education: Decorah Senior High School, Decorah, Iowa, May 1982.

Collegiate institutions attended:

8/92-1/99	University of Maryland Baltimore County Baltimore, Maryland Doctor of Philosophy, 1998.
1/86-12/86	The Johns Hopkins University Whiting School of Engineering Baltimore, Maryland Masters of Science, 1987.
7/82-5/86	The United States Naval Academy Annapolis, Maryland Bachelors of Science, 1986.

Major: Computer Science.

Professional publications:

REITAN, PAULA J. "3D Visualization of Color Image Histograms." *Computer Network and ISDN Systems*, 30(20-21):2025-35, (November 1998).

REITAN, PAULA J. AND NICHOLAS, CHARLES K. "Heterogeneous 3D Data Structure for Color Image Quantization." Poster presented at the *20th Annual Graduate Student Research Day*. University of Maryland Baltimore County, April 29, 1998. (Placed first in Electrical Engineering Poster Session.)

REITAN, PAULA J. "3D Visualization of Truecolor Image Histograms." *Sixth International Conference on Computational Graphics and Visualization Techniques*, pp. 320-9, Vilamoura, Algarve, Portugal, (December 15-18, 1997).

Professional positions:

8/98-present	Assistant Professor at the United States Naval Academy Computer Science Department United States Naval Academy Annapolis, Maryland 21402
8/92-8/98	Instructor at the United States Naval Academy Computer Science Department United States Naval Academy Annapolis, Maryland 21402
6/92-7/92	Instructor at Luther College Upward Bound Program Luther College Upward Bound Decorah, Iowa 52101.
8/91-3/92	Technical Support at Meridian Software Systems Meridian Software Systems 10 Pasteur Street Irvine, California 92715
5/86-7/91	Officer in the United States Navy Achieved rank of Lieutenant, Surface Warfare Qualified

Teaching Experience:

Courses taught at the United States Naval Academy:

SI204L	Introduction to Computer Science (lab only)	Fall 92, Fall 93
SI220	Data Structures	Spring 93, Spring 94, Fall 94, Spring 95, Fall 95, Fall 96, Spring 97
SI283	Programming for Engineers	Fall 98
SI433	Advanced Computer Algorithms	Spring 96
SI460L	Computer Graphics (lab only)	Spring 98
SI462L	Advanced Computer Graphics (lab only)	Fall 97
SI462	Advanced Computer Graphics	Fall 98

Courses taught at the University of Maryland Baltimore County:

CMSC 109	OOP in C++	Summer 95
CMSC 341	Data Structures	Summer 94

ABSTRACT

Title of Dissertation: Hybrid Approaches to Color Image Quantization

Paula Julie Reitan, Doctor of Philosophy, 1999

Dissertation directed by: Dr. Charles K. Nicholas

Associate Professor

Department of Computer Science and Electrical Engineering

Color image quantization is the irreversible transformation of a truecolor image into a color-mapped image consisting of K carefully selected representative colors. There are many possible mappings of a truecolor image to a quantized image using K colors. The primary goal of color quantization is to minimize the visual distortion between the original image and the quantized image.

This dissertation proposes a heterogeneous-cut algorithm that combines the speed of oct-cut methods with the accuracy of 24-bit precision, variance-minimization and principal component oblique-cut methods to achieve high quality quantized images quickly. This dissertation also presents a fast and effective (improves image quality) method for generalizing activity weighting to any histogram-based color quantization algorithm. The value of the heterogeneous-cut algorithm and activity weighting is validated by a comprehensive empirical analysis of thirty-nine other hierarchical color quantization techniques using a test set consisting of twenty-five diverse images. Of the high quality quantization techniques studied in the analysis, the proposed heterogeneous-cut algorithm is the fastest.

This dissertation shows that the maximum intercluster distance is not an appropriate error measure for color image quantization (MinMax). Furthermore, this dissertation proposes a new non-hierarchical color quantization technique called weighted MinMax that is a hybrid between the MinMax and Linde-Buzo-Gray (LBG) algorithms. The new

method incorporates frequency (or activity weighting) information in order to obtain high quality quantized images with significantly less visual distortion than the MinMax algorithm. However, the running time of both the MinMax and the weighted MinMax algorithm is not competitive with any of the hierarchically divisive methods.

HYBRID APPROACHES TO COLOR IMAGE QUANTIZATION

by
Paula Julie Reitan

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements of the degree of
Doctor of Philosophy
1999

DEDICATION

For those who aspire to dance like Fred,
“only backwards and in high heels.”

Ginger Rogers

ACKNOWLEDGEMENTS

I am grateful for love and support my friends and family gave me during the last very long six years. I regret that all too often my all too consuming pursuit of a Ph.D. made me miss out important events of my friends and family. I regret the same caused me not to carry my weight in family responsibilities. I owe you all more than I will ever be able to repay. Thanks for your patience and understanding.

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF COLOR PLATES	ix
1 INTRODUCTION.....	1
1.1 Image Compression.....	3
1.2 Color Quantization.....	5
1.3 Contributions.....	6
1.4 Dissertation Overview	7
2 COLOR QUANTIZATION TECHNIQUES.....	8
2.1 Truecolor Images	8
2.1.1 Truecolor Image Test Set.....	8
2.1.2 Truecolor Image Histograms.....	9
2.1.3 Image Test Set Discussion and Credits.....	13
2.2 Pre- and Post-Processing Techniques.....	16
2.2.1 Bit-cutting	16
2.2.2 Spatial Dithering.....	20
2.2.3 Subsampling	23
2.3 False Contour Reduction	24
2.3.1 Spatial Activity Weighting.....	24
2.3.2 Feedback-based Quantization	27
2.3.3 Texture Analysis.....	28
2.4 Distortion Metrics	29
2.4.1 RMSE	29
2.4.2 WRMSE.....	30

2.4.3	SNR30	
2.4.4	Maximum Diameter.....	30
2.4.5	Perceptually Uniform Color Spaces	31
2.5	Strategy for Empirical Analysis.....	32
3	HIERARCHICAL METHODS.....	34
3.1	Agglomerative.....	35
3.1.1	Fast PNN using <i>k</i> -d Trees	38
3.2	Divisive.....	39
3.2.1	Median, Center, Mean and RWM-Cut	42
3.2.2	Variance Minimization	46
3.2.3	Oct-cut	51
3.2.4	Oblique-Cut.....	51
3.3	Empirical Analysis	55
3.4	Subjective Observations	61
3.5	Summary.....	63
4	NON-HIERARCHICAL METHODS.....	70
4.1	The Popularity Algorithm.....	71
4.2	The MinMax Algorithm	73
4.2.1	Theoretical Interest	74
4.2.2	Empirical Analysis	76
4.2.3	Subjective Observations.....	77
4.3	Linde-Buzo-Gray (LBG)	79
4.3.1	Empirical Analysis	80
4.3.2	Subjective Observations.....	81
4.4	Summary.....	81
5	PROPOSED METHODS.....	83
5.1	Heterogeneous-Cut Algorithm.....	83
5.1.1	Empirical Analysis	85
5.1.2	Subjective Observations.....	88
5.2	Weighted MinMax	89
5.2.1	Empirical Analysis	91
5.2.2	Subjective Observations.....	92

5.3 Summary.....	93
6 CONCLUSIONS	94
REFERENCES.....	95

LIST OF FIGURES

Figure 1.1: Image Compression.	4
Figure 1.2: GIF Image Compression.	4
Figure 2.1: YIQ channels of Truecolor Image <i>Jhonni</i>.	19
Figure 2.2: Floyd-Steinberg Error Diffusion Filter.	21
Figure 2.3: Pseudocode for Feedback-based Color Quantization.	27
Figure 2.4: Color Quantization in Perceptually Uniform Color Spaces.	32
Figure 3.1: Pseudocode for Agglomerative Color Quantization.	36
Figure 3.2: Pseudocode for Fast PNN Color Quantization.	39
Figure 3.3: Pseudocode for Histogram-Based Divisive Techniques.	40
Figure 3.4: Pseudocode for Wu and Witten's Recursive Divisive Technique.	43
Figure 3.5: Projected Distributions for <i>Windsails</i>, $p=(8,8,8)$.	44
Figure 3.6: Activity Weighted Projected Distributions for <i>Windsails</i>, $p=(8,8,8)$.	44
Figure 3.7: Error Reduction for <i>Windsails</i>, $p=(8,8,8)$.	50
Figure 3.8: Average Running Time and RMSE of Divisive Algorithms.	58
Figure 3.9: Average Running Time and WRMSE of Activity Weighted Divisive Algorithms.	60
Figure 4.1: Pseudocode for the MinMax Algorithm.	73
Figure 4.2: Pseudocode for the LBG Algorithm.	79
Figure 5.1: Flow-chart for the Proposed Heterogeneous-Cut Algorithm.	84
Figure 5.2: Average Running Time and RMSE for Proposed Heterogeneous-Cut Algorithm.	86
Figure 5.3: Average Running Time and WRMSE for Proposed Activity Weighted Heterogeneous-cut Algorithm.	87
Figure 5.4: Pseudocode for the Weighted MinMax Algorithm.	90

LIST OF TABLES

Table 2.1: Spatial and Color Resolution of the Image Test Set.	11
Table 2.2: Bit-cutting Statistics of the Image Test Set.	18
Table 3.1: Taxonomy of Statistics Used by Divisive Algorithms.	41
Table 3.2: List of Divisive Algorithms Analyzed.	57
Table 3.3: Running Time, RMSE, and WRMSE for PNN.	59
Table 3.4: Running Time, RMSE, and WRMSE for Activity Weighted PNN.	61
Table 4.1: Average Running Time, <i>maxDiam</i>, RMSE, and WRMSE for MinMax.	77
Table 4.2: Empirical Data for LBG.	80
Table 5.1: Number of Splits Made by the Heterogeneous-Cut Algorithm.	88
Table 5.2: Empirical Data for Proposed Weighted MinMax.	91

LIST OF COLOR PLATES

Color Plate 1.1: Quantization of Truecolor Image <i>Jhonni</i>.	2
Color Plate 2.1: Image Test Set.	10
Color Plate 2.2: Histograms of the Image Test Set.	12
Color Plate 2.3: Histogram of Truecolor Image <i>Windsails</i>.	20
Color Plate 2.4: Floyd-Steinberg Error Diffusion.	22
Color Plate 2.5: Activity Weighting.	26
Color Plate 3.1: Hierarchical Clustering Trees.	34
Color Plate 3.2: Principal Components for <i>Windsails</i>.	54
Color Plate 3.3: 3D RWM for <i>Windsails</i>.	55
Color Plate 3.4: Hierarchical Quantization of <i>Solids</i>.	64
Color Plate 3.5: Hierarchical Quantization of <i>Lena</i>.	66
Color Plate 3.6: Activity Weighted Hierarchical Quantization of <i>Solids</i>.	68
Color Plate 3.7: Activity Weighted Hierarchical Quantization of <i>Lena</i>.	69
Color Plate 4.1: Popularity Quantization of <i>Solids</i> and <i>Lena</i>.	72
Color Plate 4.2: MinMax Quantization of <i>Solids</i> and <i>Lena</i>.	78
Color Plate 4.3: LBG Quantized Images.	82
Color Plate 5.1: Proposed Heterogeneous-Cut Quantization of <i>Solids</i> and <i>Lena</i>.	89
Color Plate 5.2: Proposed Weighted MinMax Quantization of <i>Solids</i> and <i>Lena</i>.	92

Chapter 1

Introduction

Color image quantization is the *irreversible* transformation of a *truecolor* image into a color-mapped image consisting of K carefully selected representative colors. When Heckbert proposed the color image quantization problem in his seminal SIGGRAPH paper [Hec82], most graphics workstations had CRT monitors with an 8-bit frame buffer, capable of displaying only 256 colors at a time. Thus, to render a truecolor image on an 8-bit frame buffer, the number of colors used to represent the image must be reduced to 256 or less. There are many possible mappings of a truecolor image to a quantized image using K colors. The primary goal of color quantization is to minimize the visual distortion between the original image and quantized image.

For example, there are noticeable differences between the truecolor image *Jhonni* and two color-mapped representations ($K=128$ and 64) displayed in Color Plate 1.1. One of the most noticeable artifacts is contouring in smooth regions such as *Jhonni*'s nose, cheeks, forehead, arm, and even the background in the lower-left corner. Contouring such as this is one of the most common problems associated with color image quantization and has been the focus of much research [BaA91a, BaAB94, ChTM94, LiC95, JoX96, KiLLH96a, KiLLH96b, Shu97]. In addition, the specular highlight on her nose is objectionably noticeable. A less noticeable artifact present in Color Plate 1.1 is a reddish shift in the color of *Jhonni*'s hair. From a subjective point of view, the quantized images in Color Plate 1.1 are not acceptable since they contain quantization artifacts that are visually objectionable.



Color Plate 1.1: **Quantization of Truecolor Image *Jhonni*.** (a) *Jhonni* has 240,000 pixels (600x400) and 67,085 unique colors in RGB space. *Jhonni* has been quantized to (b) $K=128$ and (c) $K=64$ colors in RGB space. Images are printed at 65 percent of their original spatial resolution. *Jhonni* is courtesy of Dr. Andrew T. Phillips.

1.1 Image Compression

With the advent of the multimedia era, 24-bit frame buffers have become commonplace. However, researchers continue to be interested in the color image quantization problem because 8-bit frame buffers remain in existence and because of this problem's applicability to image compression. As the multimedia and Internet eras progress at full steam ahead, both the demand and supply of images are increasing at an alarming rate; thus the need for efficient ways of representing images has become increasingly more important.

Images are typically compressed to reduce the amount of data required to represent an image. All image compression techniques embody both a *compression* algorithm and a *reconstruction* algorithm. The image compression process is illustrated in Figure 1.1. If an image compression technique guarantees that, for any image I_C , the reconstructed image I_R , will be identical to the original image I , then it is called *lossless*; otherwise it is called *lossy*. As such, color image quantization is a form of lossy image compression.

Color quantization is a fundamental component of several lossy image compression formats, most notable is the Graphics Interchange Format (GIF) which is a common image format found on the Internet. GIF compresses colormapped images using a variant of the Lempel Ziv Welch (LZW)¹ dictionary-based coding technique [MuV94 and Say96]. The GIF image compression technique is illustrated in Figure 1.2. GIF requires that a truecolor image first be color quantized to 256 or fewer colors. The GIF format first stores the colormap, then the image data. Each pixel is encoded by the index of its representative color in the colormap. LZW builds a dictionary of patterns; the patterns may be repeated using a reference to its dictionary entry. LZW is a lossless image compression technique and works particularly well for computer generated images which often contain large blocks of the same color. Since color quantization is the only lossy part of compressing a truecolor image using GIF, the color quantizer, Q , directly impacts the fidelity of the compressed image. Therefore improvements in color quantization techniques remain of great value.

¹ LZW is patented by Unisys, United States Patent No. 4,558,302.

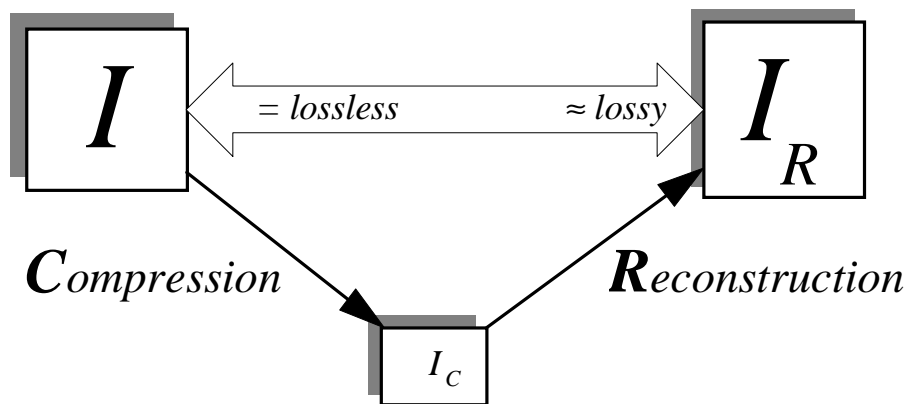


Figure 1.1: **Image Compression.** The compression algorithm encodes an image, I , into a smaller image, I_C , such that the reconstruction algorithm can decode I_C to I_R , where I_R is either identical to I or is at least a reasonably close approximation to I .

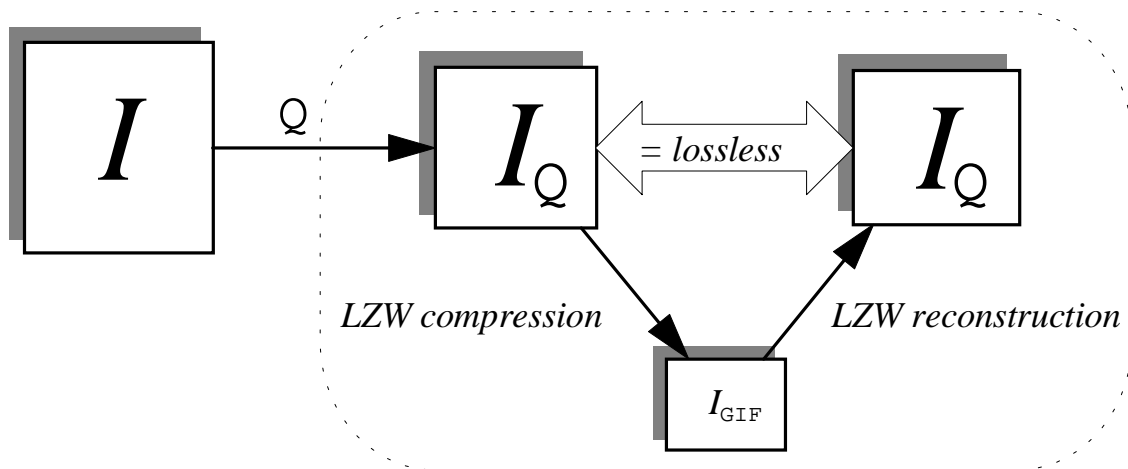


Figure 1.2: **GIF Image Compression.** First a truecolor image, I , is color quantized using the color quantizer Q to a color-mapped image I_Q . The color-mapped image is then compressed using LZW to a GIF image I_{GIF} . The LZW reconstruction algorithm faithfully restores the color-mapped image I_Q from I_{GIF} . Note that due to the color quantization $I_Q \neq I$.

1.2 Color Quantization

One of the primary tasks of color quantization is to select a set, R , of K colors:

$$R = \{ \mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{K-1} \} \subset \text{RGB} \quad (1.1)$$

to represent the unique colors in the truecolor image. The algorithm used to select the K representative colors will be generically referred to as `Select`. Once the K representative colors have been selected, each of the pixels in the truecolor image is mapped to a representative color usually chosen to minimize a given distortion metric (Section 2.4). The algorithm used to map image pixels to a representative color will be called `Map`. Collectively, `Select` and `Map` will be referred to as the color quantizer Q .

Numerous and varied techniques have been proposed for the selection of representatives. Uniform color quantization techniques use the same set of representative colors for all truecolor images. On the other hand, adaptive color quantization techniques select a set of representative colors for each truecolor image. Since uniform quantization techniques generally produce poor quality color-mapped images, this dissertation only describes and compares adaptive techniques. This dissertation organizes techniques for `Select` into two broad categories:

- (i) *hierarchical* (Chapter 3) and
- (ii) *non-hierarchical* (Chapter 4).

This form of classification is commonly used in clustering analysis that strives to discover structure in complex data. Since many color quantization techniques were first proposed as general-purpose clustering algorithms this classification scheme is natural. Hierarchical methods use divisive (or agglomerative) techniques to build a tree from the top down (or bottom up) containing the set of unique colors in the truecolor image. Non-hierarchical techniques include methods that use iteration, self-organizing neural networks, and genetic programming to refine previously selected representatives.

Let C be the set of unique colors in a truecolor image. Map induces a partitioning

$$S = \{s_0, s_1, \dots, s_{K-1}\} \quad (1.2)$$

of C into K clusters called a K -split. Each cluster s_i contains the colors in the truecolor image which are mapped to the representative color r_i .

Various methods have been proposed for mapping image colors to representative colors. The most accurate, but slowest method is to map each color to its nearest neighbor. Heckbert proposed using a specialized data structure to preprocess the image colors into a locally sorted list, thus limiting the number of possible nearest neighbors [Hec82]. Spatial subdivisions such as k -d trees and octrees can also be used to speed up the search for a representative color. Centroid mapping [RoG95] is applicable when the technique used to select the representatives results in a spatial subdivision having K leaves. Centroid mapping is very fast because the centroid of each leaf represents all the colors in the leaf's domain; however, it is not as accurate as using the nearest neighbor.

1.3 Contributions

This dissertation proposes a heterogeneous-cut algorithm that combines five of these divisive algorithms in a unique way that capitalizes on the strengths of each technique, but does not suffer from their weaknesses. In summary, this dissertation makes the following contributions:

- 1) Section 2.3.1 proposes a fast and effective (improves image quality) method for generalizing any histogram-based non-activity weighted color quantization algorithm to a histogram-based, activity weighted color quantization algorithm.
- 2) Section 3.2 introduces a general taxonomy for classifying hierarchically divisive color quantization methods.
- 3) Chapter 3 provides a comprehensive survey and comparative analysis of thirty-nine hierarchical color quantization methods using a test set consisting of twenty-five diverse images. Some of the methods discussed in this chapter have not been previously studied.

- 4) Section 4.2 shows that the maximum intercluster distance is not an appropriate error measure for color image quantization (MinMax).
- 5) Section 5.1 proposes an adaptive heterogeneous-cut algorithm that enables fast, but high quality hierarchically divisive quantization in full 24-bit precision. Of the high quality quantization techniques studied in the dissertation, the proposed heterogeneous-cut algorithm is the fastest.
- 6) Section 5.2 proposes a new non-hierarchical color quantization technique called weighted MinMax that is a hybrid between the MinMax and Linde-Buzo-Gray (LBG) algorithms. The new method incorporates frequency (or activity weighting) information in order to obtain high quality quantized images with significantly less visual distortion than the MinMax algorithm. However, the running time of both the MinMax and the weighted MinMax algorithm is not competitive with any of the hierarchically divisive methods.

1.4 Dissertation Overview

This section describes the overall organization of the dissertation. The next chapter describes techniques that are generally applicable to many color quantization methods. The chapter concludes with a description of the strategy used to perform the empirical analyses in Chapters 3-5. Specifically, Chapter 3 surveys and analyzes hierarchical quantization methods, while Chapter 4 surveys and analyzes some of the non-hierarchical color quantization methods. The proposed heterogeneous-cut algorithm described in Section 5.1 is motivated by the empirical analysis presented in Section 3.3. The heterogeneous-cut algorithm enables fast, but high quality hierarchically divisive quantization in full 24-bit precision. Chapter 5 also proposes the weighted MinMax algorithm that is a hybrid between MinMax and LBG. Section 5.2 shows that the weighted MinMax algorithm produces quantized images with excellent quality (far superior to MinMax).

Chapter 2

Color Quantization Techniques

2.1 Truecolor Images

Stokes, Fairchild and Berns [StFB92] experimentally showed that a minimum of 7.4-bits of precision are required in each of the red, green, and blue color directions to precisely display full-color images. In this dissertation the full-color images are stored in Targa's Truecolor image format [MuV94] which represents each image pixel using 8-bits of precision for each red, green, and blue channel:

$$\text{RGB} = \{ (r, g, b) \mid r, g, b \in \mathbb{Z}_{256} \}. \quad (2.1)$$

Thus, RGB denotes the set of 256^3 (16M) possible colors in a truecolor image. A truecolor image I is more formally defined as a total function:

$$I: \mathbb{Z}_H \times \mathbb{Z}_W \rightarrow C, \quad (2.2)$$

where $I(x, y)$ is the RGB color of the pixel at row x , column y of image I and

$$C = \{ c_1, c_2, \dots, c_N \} \subseteq \text{RGB}. \quad (2.3)$$

Thus, C is the set of N unique colors present in image I .

2.1.1 Truecolor Image Test Set

Twenty-five natural and computer generated color images were selected as an "in practice average-case" test set of truecolor images. Thumbnails of the images in the test set are displayed in Color Plate 2.1. Copyright issues make the compilation of a large, diverse and comprehensive image test set difficult; nonetheless, the test set used in this

dissertation contains a good variety of images typically encountered in practice. Table 2.1 shows there is a wide range in the number of pixels and in the number of unique colors in the images of the test set.

The test set images were selected because they exhibit many of the characteristics commonly found in color digital imagery:

- Low frequency (smoothly varying) regions where false contours are more noticeable,
- Isolated colors which force color shifting, and
- High frequency regions where sharpness may be deteriorated.

Section 2.1.3 provides specific rationale for including each of the images in the test set.

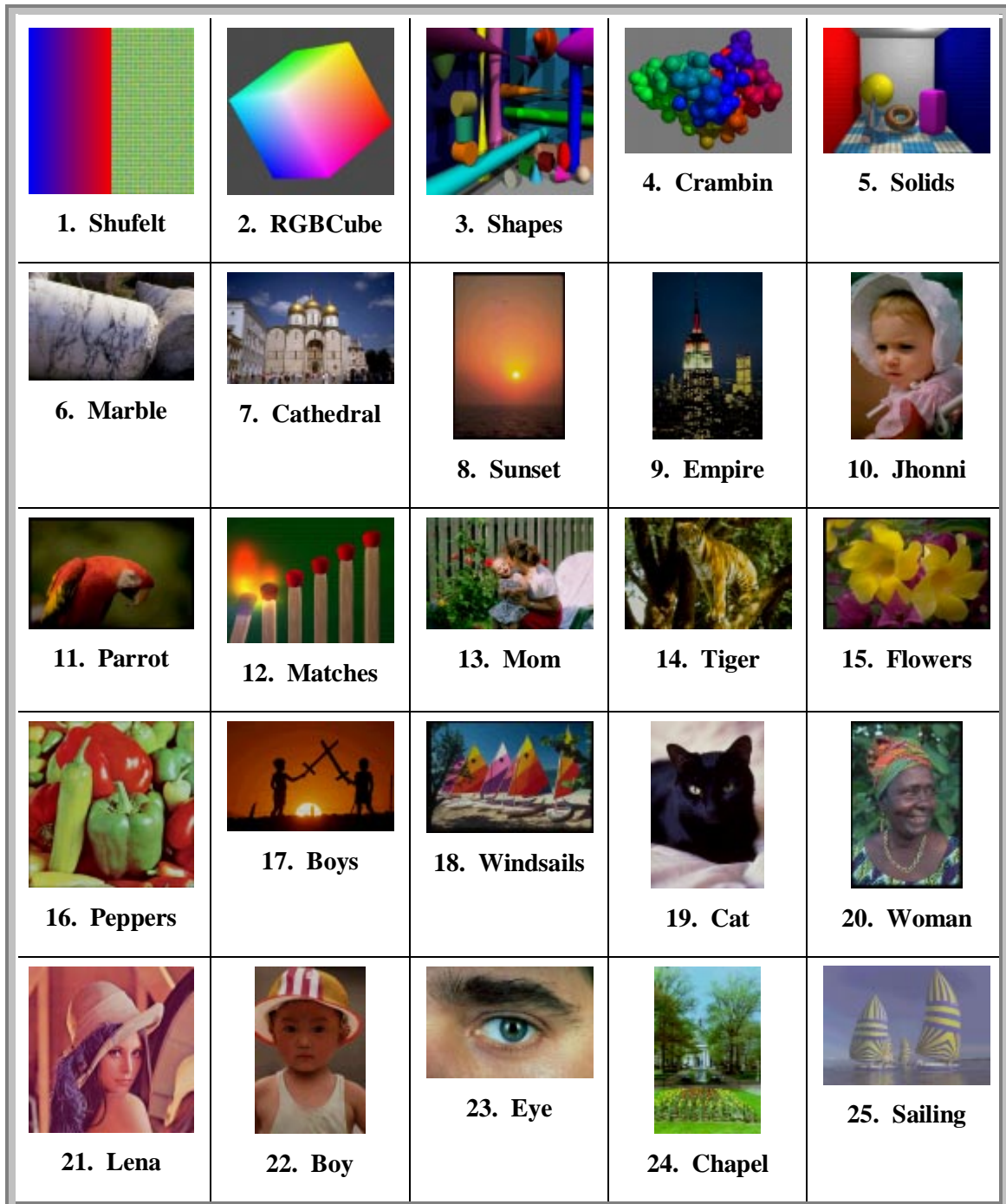
2.1.2 Truecolor Image Histograms

The first step of many color quantization techniques is to create a histogram of the colors in the truecolor image. The histogram H of a truecolor image I is defined as a total function:

$$H: \text{RGB} \rightarrow \mathbb{N} \tag{2.4}$$

where $H(c)$ is the number of pixels in the image I with color c . Note that H does not contain any spatial information about the pixels in I .

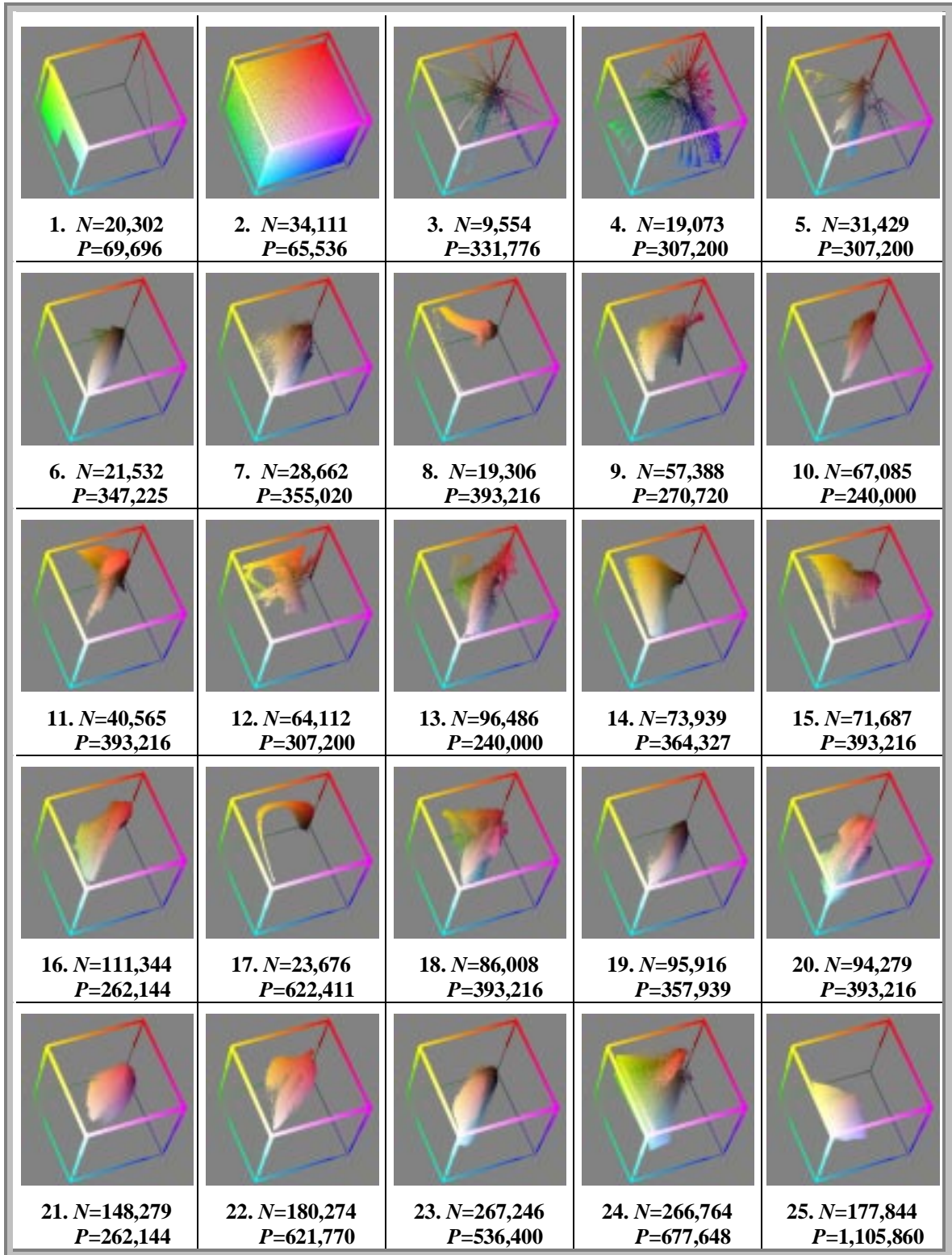
Color Plate 2.2 displays a view of the histogram for each image in the test set. These histograms show that there is a good variety in the shape of the image test set histograms. The histogram visualizations presented in this dissertation were created using a C++ program developed by the author called LindyHop. Various other methods for visualizing truecolor image histograms are described in [Rei97 and Rei98].



Color Plate 2.1: **Image Test Set.** Twenty-five truecolor images selected to analyze the accuracy and time requirements of the color image quantization techniques discussed in this dissertation

#	Name	W	H	$P = W \cdot H$	N
1	<i>Shufelt</i>	264	264	69,696	20,302
2	<i>RGBCube</i>	256	256	65,536	34,111
3	<i>Shapes</i>	576	576	331,776	9,554
4	<i>Crambin</i>	640	480	307,200	19,073
5	<i>Solids</i>	640	480	307,200	31,429
6	<i>Marble</i>	731	475	347,225	21,532
7	<i>Cathedral</i>	732	485	355,020	28,662
8	<i>Sunset</i>	512	768	393,216	19,306
9	<i>Empire</i>	423	640	270,720	57,388
10	<i>Jhonni</i>	400	600	240,000	67,085
11	<i>Parrot</i>	768	512	393,216	40,565
12	<i>Matches</i>	640	480	307,200	64,112
13	<i>Mom</i>	600	400	240,000	96,486
14	<i>Tiger</i>	739	493	364,327	73,939
15	<i>Flowers</i>	768	512	393,216	71,687
16	<i>Peppers</i>	512	512	262,144	111,344
17	<i>Boys</i>	971	641	622,411	23,676
18	<i>Windsails</i>	768	512	393,216	86,008
19	<i>Cat</i>	491	729	357,939	95,916
20	<i>Woman</i>	512	768	393,216	94,279
21	<i>Lena</i>	512	512	262,144	148,279
22	<i>Boy</i>	641	970	621,770	180,274
23	<i>Eye</i>	900	596	536,400	267,246
24	<i>Chapel</i>	652	1,024	677,648	266,764
25	<i>Sailing</i>	1,220	863	1,052,860	177,844
	Max	1,220	1,024	1,052,860	267,246
	Min	256	256	65,536	9,554
	Avg	635	582	382,212	84,274

Table 2.1: **Spatial and Color Resolution of the Image Test Set.** P is the number of pixels in the image; N is the number of unique RGB colors in the image.



Color Plate 2.2: **Histograms of the Image Test Set.** Snap shot of the histogram of each image in the test set.

2.1.3 Image Test Set Discussion and Credits

A description of each of the twenty-five images in the test set is provided below.

1. *Shufelt* was created using Shufelt's description of his first test image in [Shu97]. This image was designed to contain a large smooth area in which contouring can be a problem as well as a textured area in which large quantization errors may exist without being noticed by the observer.
2. *RGBCube* was designed to contain large, smoothly varying region in which contouring is problematic. It was modeled after an image used by Xiang [Xia97] to test the MinMax quantization technique (Section 4.2).
3. *Shapes* was created using the Persistence of Vision Raytracer version 3.0 (POV-Ray) located on the web at (<http://www.povray.org>). The scene description is located in povray3/pov3demo/demo/shapes.pov. *Shapes* was included to contain large, smoothly varying areas as well as regions with sharp edges.
4. *Crambin* was created using PovChem version 1.0 by Paul Thiessen (<http://cherubino.med.jhmi.edu/~paul/PovChem.html>) and POV-Ray. The molecule rendered is called crambin, and its description (crambin.pdb) was provided by Dr. Andrew T. Phillips.
5. *Solids* was also created using POV-Ray. The scene description is a slightly modified version of povray3/pov3demo/radios/rad1.pov. *Solids* was included to contain large, smoothly varying areas as well as sharp edges.
6. *Marble* is image israel\1558_050.jpg from the *World Photo* CD-ROM (Aztech New Media Corporation, <http://www.aztech.com>). It was included because it contains natural textures.
7. *Cathedral* is image russia\1548_068.jpg also from the *World Photo* CD-ROM. It was included because skies are particularly susceptible to contouring. Also, the gold in the dome is a prominent feature of the image, but is isolated from the other colors in the image.

8. *Sunset* is image Photo_cd\Img0015.pcd from the *Textures* CD-ROM (Arc Media Incorporated, Buffalo, New York, 1994). It was selected because the smoothly varying region of yellows, oranges and reds around the setting sun are particularly susceptible to contouring.
9. *Empire* is image misc\8000_001.jpg also from the *World Photo* CD-ROM. It was selected because of the high contrast in colors from the darkness of the night and the lights of the city.
10. *Jhonni* is courtesy of Dr. Andrew T. Phillips. Close-ups of people are particularly difficult to quantize, because we have very strong expectations of what a human face should look like. Even the smallest amount of error may be quite noticeable and objectionable when located in a close-ups people.
11. *Parrot* is image Photo_cd\Img0077.pcd from also from the *Textures* CD-ROM. It was selected because it resembles a test image used by Velho, Gomes and Sobreiro [VeGS97] to test the pairwise nearest neighbor quantization technique (Section 3.1).
12. *Matches* was also created using POV-Ray. The scene description is located in povray3/pov3demo/showoff/matches.pov. *Matches* was selected because it is a synthetic image which contains many colors. The area around the lit match contains many of the same characteristics found in *Sunset* and *Boys*.
13. *Mom* is also courtesy of Dr. Andrew T. Phillips. It was included because it contains quite a large number of colors that are spread nicely throughout the RGB space.
14. *Tiger* is image india\1071_031.jpg from the *World Photo* CD-ROM. It was included because it contains natural textures.
15. *Flowers* is image Photo_cd\Img0081.pcd also from the *Textures* CD-ROM. It was selected because the smoothly varying regions of yellow on the petals are particularly difficult to quantize.
16. *Peppers* is from the University of Waterloo's ColorSet Repertoire (<http://links.uwaterloo.ca/colorset.base.html>). *Peppers* was included because it is a classic image used in image compression and image quantization research. The

specular highlights on the peppers come from a small area isolated from the rest of the colors in the image; thus they tend to become objectionably noticeable.

17. *Boys* is image Learning\Boys.lrg from the *World Photo* CD-ROM. It was included because the shape of its histogram is very similar to *Sunset's*; however, its spatial resolution is about twice that of *Sunset*.
18. *Windsails* is image 68063 from the *Caribbean Professional Photo* CD-ROM, (Corel, Ottawa, Canada, 1994). It was selected because it contains a smoothly varying sky as well as many colors widely distributed in the RGB space.
19. *Cat* is image canada\0563_032.jpg from the *World Photo* CD-ROM. It was selected because it contains natural textures and because the sharp contrast between the black fur and the gold eyes may be problematic. In addition, the image is of interest because its histogram lies very close to greyscale.
20. *Woman* is image 68014 also from the *Caribbean Professional Photo* CD-ROM. It was selected to determine the difficulty of quantizing people of color versus Caucasians (*Lena* and *Jhonni*).
21. *Lena* is also from the University of Waterloo's ColorSet Repertoire. Like *Peppers*, *Lena* is a classic image used in image compression and image quantization research. *Lena* has many of the same characteristics found in *Jhonni*.
22. *Boy* is image Images\Photodisc\Med_res\Ss03085.lrg from the *World Photo* CD-ROM. Like *Woman*, *Boy* was selected to add some dimensionality to the color of the people in the test set.
23. *Eye* is image Images\Imagekit\People\09ple103.pct from the *World Photo* CD-ROM. It was included because the eye contains both smoothly varying and high contrast areas.
24. *Chapel* was taken by Ken Mierzejewski and is courtesy of the United States Naval Academy Public Affairs Office. It was selected because it has large spatial and color resolutions.

25. *Sailing* was taken by David Eckard and is also courtesy of the United States Naval Academy Public Affairs Office. Like *Chapel*, it was selected because it has large spatial and color resolutions.

2.2 Pre- and Post-Processing Techniques

This section describes pre- and post-processing techniques that are applicable to any color quantization technique. Bit-cutting and subsampling are pre-processing techniques used to speed-up algorithms generally at the expense of image quality. Spatial dithering is a post-processing technique that generally enhances image quality with a moderate cost of time.

2.2.1 Bit-cutting

Heckbert suggested cutting the three least-significant bits from each R, G, B component [Hec80]. Bit-cutting in this manner uniformly quantizes the RGB color space into a smaller color space containing 32K (32^3) possible colors. Most histogram-based color quantization techniques bit-cut the image colors prior to inserting them into the histogram, thus saving space and time at the expense of color accuracy.

Joy and Xiang [JoX93] showed that bit-cutting non-uniformly (three bits from the red component, two bits from the green component, and four bits from the blue component) can result in less visual distortion than uniformly cutting three bits from each component when using the center-cut quantization technique (Section 3.2.1). Joy and Xiang's non-uniform bit-cutting also results in a smaller color space containing 32K ($32 \times 64 \times 16$) possible colors. This non-uniform bit-cutting corresponds approximately to the luminance component, Y , of the YIQ model:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.5)$$

Row one of Equation 2.5 shows that the relative weights of R , G , and B for the Y component are 0.299:0.587:0.114 which is approximated by the 32:64:16 ratio achieved by the non-uniform bit-cutting proposed by Joy and Xiang.

YIQ is used in the United States for commercial color television transmission where only the Y component is shown on black-and-white televisions. In contrast, color televisions display the luminance channel as well as the chrominance channels (I and Q). The mere existence of black-and-white televisions is testament to the vast amount of information carried by the luminance channel of an image. Figure 2.1 shows the Y , I and Q channels of *Jhonni*. Notice how most of the fine details and depth information of *Jhonni* is captured by the luminance channel displayed in Figure 2.1.(a).

Row one of Equation 2.5 shows that green and red are relatively important and blue is relatively unimportant to the brightness of a color. Joy and Xiang's non-uniform bit-cutting exploits the following property of the human visual system: the human eye is more sensitive to shifts in luminance than in chrominance. Non-uniform bit-cutting mimics the human observer's sensitivity to luminance and places greater importance on the luminance than on the chrominance of an image.

The number of bits of precision considered in each of the R , G , and B directions respectively will be denoted by a triple:

$$p = (r, g, b). \quad (2.6)$$

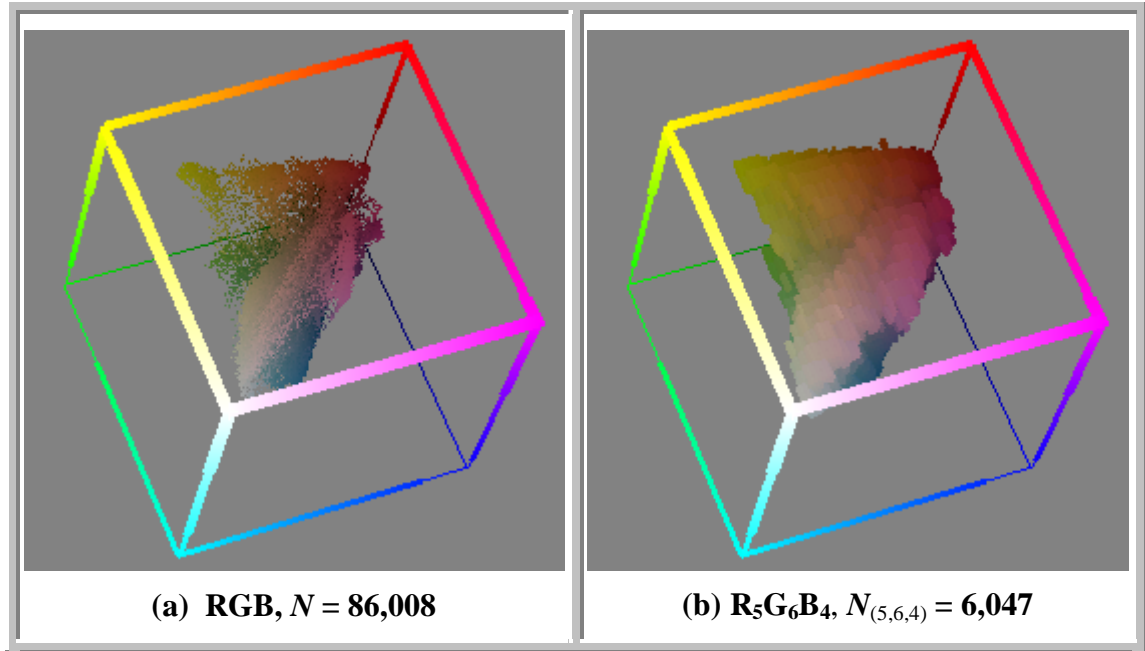
By cutting the $(8-p.\{r,g,b\})$ least significant bits from each R , G , B component respectively, the RGB color space is quantized to a smaller color space denoted as $R_{p,r}G_{p,g}B_{p,b}$. The number of unique $R_{p,r}G_{p,g}B_{p,b}$ colors in a truecolor image will be denoted as $N_{(p,r, p.g, p.b)}$. When N is not subscripted, full 24-bit precision is assumed; that is $N = N_{(8,8,8)}$. Table 2.2 shows how bit-cutting reduces the number of unique colors in a truecolor image, thus saving space and time at the expense of accuracy.

	$N_{(8,8,8)}$	$N_{(7,8,6)}$	$N_{(7,7,7)}$	$N_{(6,7,5)}$	$N_{(6,6,6)}$	$N_{(5,6,4)}$	$N_{(5,5,5)}$
1	20,302	6,207	10,906	1,628	3,188	431	828
2	34,111	30,477	31,810	13,998	12,100	3,474	2,978
3	9,554	7,145	6,408	4,477	4,120	2,278	2,102
4	19,073	12,582	11,787	6,754	6,416	3,013	2,860
5	31,429	17,580	16,866	6,586	6,295	2,070	1,962
6	21,532	15,654	17,163	6,159	6,465	1,495	1,535
7	28,662	23,075	25,273	11,188	11,767	3,133	3,149
8	19,306	11,649	12,351	3,560	3,436	853	810
9	57,388	33,594	33,507	14,687	14,616	4,067	3,945
10	67,085	46,399	46,735	13,161	12,779	2,578	2,460
11	40,565	29,412	28,408	11,399	10,906	2,746	2,576
12	64,112	30,109	27,584	10,479	9,239	3,005	2,645
13	96,486	70,748	71,365	25,257	25,025	6,324	6,218
14	73,939	52,078	53,246	17,812	17,329	3,442	3,480
15	71,687	51,897	53,356	17,352	17,158	3,576	3,527
16	111,344	79,187	79,098	23,755	23,235	4,712	4,564
17	23,676	7,778	6,673	2,539	1,716	788	492
18	86,008	68,168	72,663	26,704	26,233	6,047	5,873
19	95,916	29,924	29,889	7,693	7,628	1,877	1,881
20	94,279	73,614	78,059	29,739	29,458	6,738	6,491
21	148,279	54,063	53,812	13,139	13,018	2,650	2,596
22	180,274	58,942	58,285	15,733	15,336	3,314	3,206
23	267,246	84,531	84,390	19,536	19,436	3,758	3,694
24	266,764	145,488	140,071	38,211	36,748	8,110	7,648
25	177,844	57,592	58,213	13,313	13,296	2,389	2,368
Max	267,246	145,488	140,071	38,211	36,748	8,110	7,648
Min	9,554	6,207	6,408	1,628	1,716	431	492
Avg	84,274	43,916	44,317	14,194	13,878	3,315	3,196

Table 2.2: **Bit-cutting Statistics of the Image Test Set.**



Figure 2.1: **YIQ channels of Tricolor Image *Jhonni*.** (a) Luminance channel Y and chromaticity channels (b) I and (c) Q of tricolor image *Jhonni*.



Color Plate 2.3: **Histogram of Truecolor Image *Windsails*.** Two 3D visualizations of the *Windsails* histogram in (a) 24-bits and (b) 15-bits of precision.

Color Plate 2.3 shows two 3D visualizations of the *Windsails* histogram. Notice that the number of unique colors stored in the 15-bit histogram (Color Plate 2.3.b) is a small fraction (0.0703) of that stored in the 24-bit histogram (Color Plate 2.3.a).

Many color quantization researchers find 15-bits or 18-bits of precision ($R_5G_5B_5$, $R_5G_6B_4$, $R_6G_6B_6$, or $R_6G_7B_5$) to be a satisfactory tradeoff between space, time, and accuracy. However, Shufelt [Shu95] showed that working with less than 21-bits of precision can noticeably impair the quality of the quantized images. Thus, a major contribution made by this dissertation is a fast color quantization technique for full 24-bit precision.

2.2.2 Spatial Dithering

Spatial dithering is a remarkable technique that can dramatically enhance the perceptual quality of color-quantized images [FlS76, JaJN76, Knu86, Uli88, and GeRS93]. Dithering is also referred to as digital halftoning: a process commonly used by newspapers to print greyscale pictures using only black and white dots. In the case of

color images, dithering is used to display truecolor images using pixels consisting of K different colors. There are two basic ways of incorporating dithering into a color quantization algorithm:

- (i) *modulate* the truecolor image with high frequency noise prior to quantization, or
- (ii) *disperse* the quantization error to neighboring pixels.

Floyd-Steinberg error diffusion is the most common form of dithering used by color quantization algorithms and is quite simple to implement. The image is quantized from left to right, top to bottom. The error (Section 2.4) between the original pixel color and the quantized pixel color is dispersed to neighboring pixels not yet quantized using the proportions given in Figure 2.2. Dithering usually leads to images that appear to contain less error because the color error has been spatially dispersed as high frequency noise that is less perceptible to the human observer. Thus the eye's powerful spatial integration of the colors and the noise contained in a neighborhood of pixels creates the illusion of more colors and tends to smooth the contouring artifacts found in color quantized images.

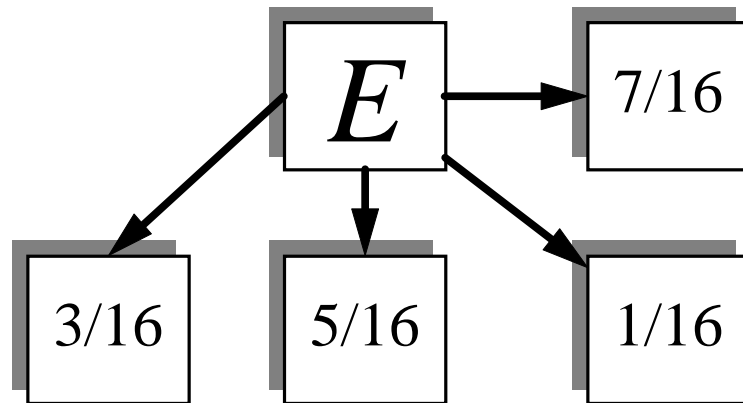
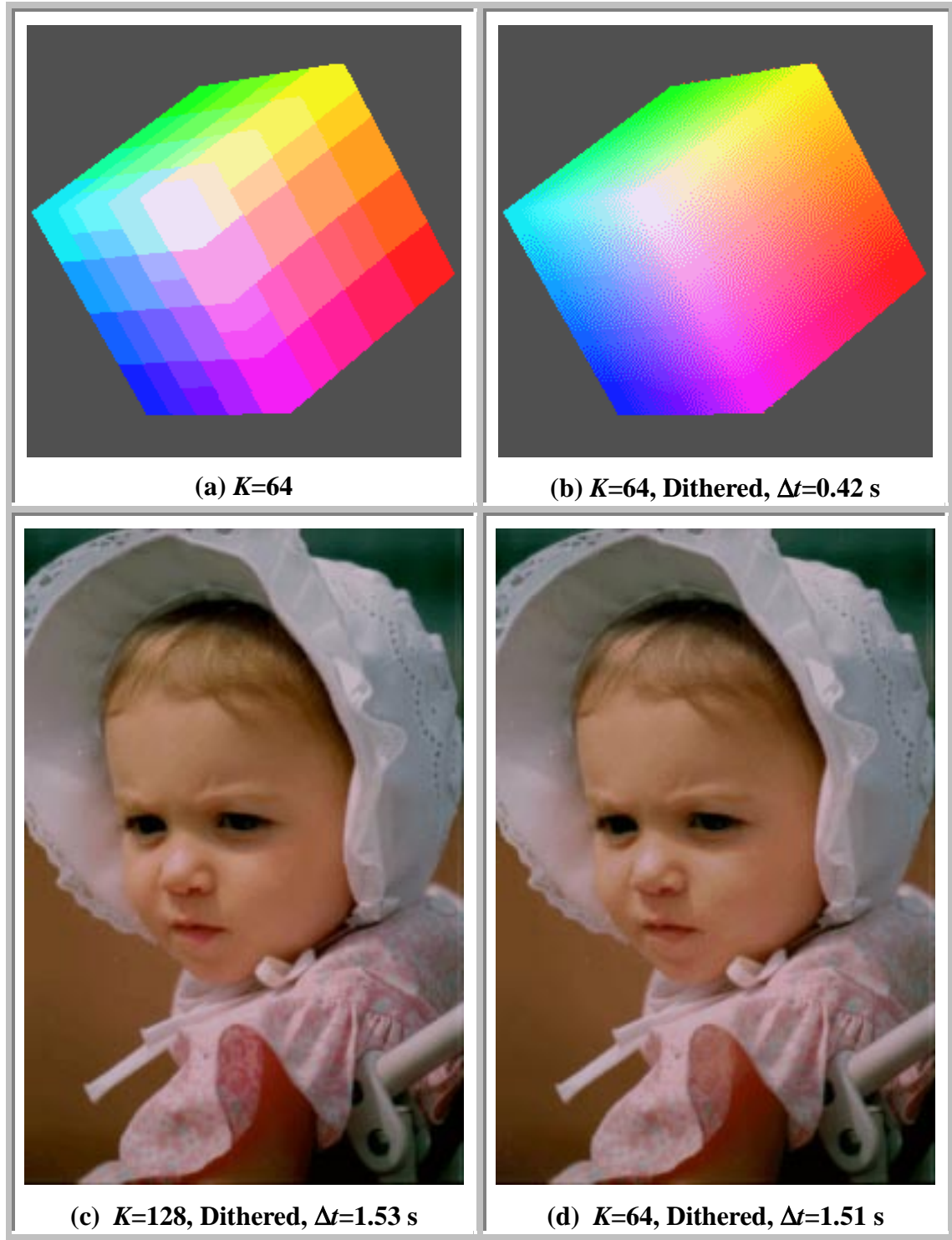


Figure 2.2: **Floyd-Steinberg Error Diffusion Filter.** E represents the error between $I(x, y)$ and $I_Q(x, y)$. E is dispersed to neighboring pixels using the proportions given in the boxes.



Color Plate 2.4: **Floyd-Steinberg Error Diffusion.** *RGBCube* is (a) quantized to $K=64$ colors and (b) dithered. The two quantized images of *Jhonni* in Color Plate 1.1.(b and c) have been dithered in (c) and (d), respectively. Δt is the additional amount of time required to perform the dithering.

Color Plate 2.4.(a and b) beautifully illustrate the power of error diffusion. There is an inordinate amount of contouring visible when *RGBCube* is quantized to 64 colors. Floyd-Steinberg error diffusion smoothes the contouring resulting in an image which is visually quite acceptable. Note that each image contains only 64 colors, but that the dithered image gives the illusion of many, many more. The snake-like patterns visible in Color Plate 2.4.b are the major drawback of Floyd-Steinberg error diffusion. Color Plate 2.4.(c and d) show how the contouring in Color Plate 1.1.(b and c) can be nearly eliminated using Floyd-Steinberg error diffusion. Liu and Chang [LiC95] proposed using the morphological erosion operator to extract boundary pixels and to thereby selectively apply error diffusion only where abrupt changes in color occur.

The time required to perform Floyd-Steinberg error diffusion is proportional to the number of pixels in the truecolor image; Δt in Color Plate 2.4 is an empirical measurement of this quantity. Note that the additional amount of time required to dither *Jhonni* is about 3.6 times that of *RGBCube*, and that this corresponds to the observation that the spatial resolution of *Jhonni* is about 3.7 times that of *RGBCube*.

Since dithering can dramatically improve the quality of quantized images, several researchers have looked at color quantization in a slightly different way. Instead of selecting the K best colors to directly represent the truecolor image, some algorithms select the K best dither colors to represent the truecolor image [HoD86, Gol91, LeTT96, AkOY96, AkOA97, and ScD97]. The key modification promoted by the quantizers designed specifically for dithering is to enlarge the convex hull of the dither colors to provide a larger color domain for the ditherer to blend colors.

2.2.3 Subsampling

When building the histogram of a truecolor image, often one can obtain the “essence” of the image without examining every pixel. That is, one can use a subset of the image pixels called the *training set*, to build the histogram. Many of the non-hierarchical methods are extremely slow, and subsampling is one method commonly used to speed up these algorithms. The training set may be chosen randomly yielding a nondeterministic algorithm. Dixit [Dix91] randomly samples a fixed number of pixels (1024) to implement a fast pairwise nearest neighbor (PNN) algorithm (Section 3.1). Verevka and

Buchanan [VeB95] advocate sampling the image at decreasing step sizes (1009,757,499,421,307,197,...) when refining their representative colors using the Local K -means algorithm. They report very good results by sampling only about 10% of the pixels in the image. Another common method for obtaining the training set is to examine every n th pixel in every m th row. For example, Goldberg [Gol91] found that sampling one fourth the image (by examining every other pixel in every other row) was sufficient for their variation on the MinMax algorithm (Section The MinMax Algorithm).

2.3 False Contour Reduction

It is well known that the human visual system is more sensitive to errors in low activity (smooth) regions than to errors in high activity (busy) regions. Thus the premise of the techniques discussed in this section is to move quantization error from low frequency areas to high frequency areas so that the error will be less visible to the human eye.

2.3.1 Spatial Activity Weighting

Various methods for weighting the importance of a pixel based on the spatial activity of the area of the image where the pixel appears have been incorporated into color quantization algorithms. In all cases, the pixel's importance (activity weighting) is inversely proportional to the activity level of the region where it appears in the image. Pixel-based methods store an activity weighting for each pixel in the image [OrB91, ChTM94, KiLLH96a, KiLLH96b]. Histogram-based methods must combine the activity weightings of all pixels with the same color. Some researchers average the activity weightings [BaAB94, PaLKLH96], while others use the maximum [BaA91a]. This dissertation proposes using the sum. Some methods speed the calculation of activity levels by computing an activity level for each (4x4 or 8x8) spatial block of the image and then assigning the same level to all pixels in the block [BaAB94, ChTM94, KiLLH96a, KiLLH96b].

This dissertation calculates the activity level of a pixel in a manner similar to Balasubramanian et al. [BaAB94]. The activity level $a(x,y)$ of a pixel at row x , column y

of a truecolor image I is measured by the magnitude of the luminance gradient of the pixel given by:

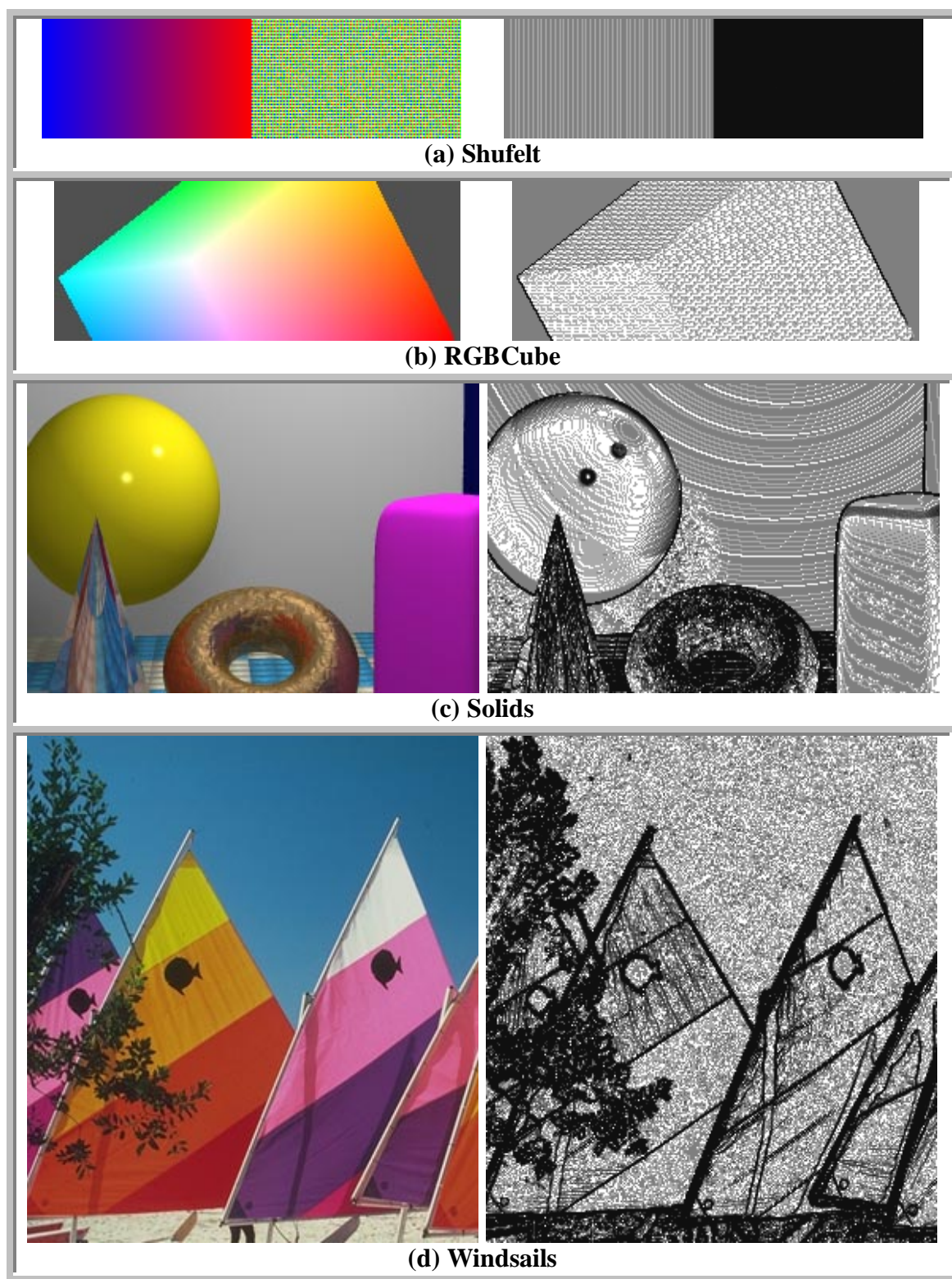
$$a(x,y) = | Y(I(x,y)) - Y(I(x-1,y)) | + | Y(I(x,y)) - Y(I(x,y+1)) |, \quad (2.7)$$

where Y is the luminance of the pixel calculated using row one of Equation 2.5, and assuming $x > 0$ and $y < H-1$. When x is 0, the first term of Equation 2.7 is $| Y(I(x,y)) - Y(I(x+1,y)) |$ and when y is $H-1$, the second term of Equation 2.7 is $| Y(I(x,y)) - Y(I(x,y-1)) |$. This dissertation calculates the activity weighting $w(x,y)$ of a pixel at row x , column y of a truecolor image using:

$$w(x,y) = \begin{cases} 1/4 & \text{if } a(x,y) = 0 \\ 1/3 & \text{if } a(x,y) = 1 \\ 1/a(x,y) & 1 < a(x,y) < 12 \\ 1/a(x,y)^{1.25} & 12 \leq a(x,y) \leq 16 \\ 1/16^{1.25} & \text{if } a(x,y) > 16 \end{cases} \quad (2.8)$$

The author conducted a considerable amount of experimentation with various exponents and cut-off values to derive Equation 2.8. Section 3.3 shows that Equation 2.8 does a good job of accurately weighting the relative importance of a pixel based on its activity level. When the activity level is extremely low (0 or 1) the pixel is considered to be part of a flat region; therefore, it is considered less important than when its activity level is low (2). When the activity level is greater than 16, the pixel is considered to be part of an edge. Thus to prevent the blurring of edges, the activity weighting of a pixel is never less than $1/16^{1.25}$.

Color Plate 2.5 illustrates the activity weighting for truecolor images *Shufelt*, *RGBCube*, *Solids*, and *Windsails*. Notice how the smooth regions are given more weight (lighter) than the textured (darker) regions, thus enabling quantization errors to be hidden in areas where the error will be less noticeable to the human eye. Unfortunately, in images such as *RGBCube*, all of the non-background pixels are weighted approximately equally very important because they all belong to a very smoothly varying region. Thus, activity weighting does little to improve quality of the quantized image. Images of this nature are very difficult to quantize; however, dithering is an excellent option for improving the visual quality of the quantized images.



Color Plate 2.5: **Activity Weighting.** Activity weighting of (a) *Shufelt*, (b) *RGBCube*, (c) *Solids* and (d) *Windsails*. The intensity of each pixel is scaled by the activity weighting of the pixel.

In histogram-based, non-activity weighted algorithms, the importance of a color is based solely upon its frequency. That is, colors that occur more often in the image are given higher importance. This dissertation proposes to weight the importance of a color based upon its frequency and the activity level of the locations where the pixels occur by summing the activity weightings of pixels with the same color. Thus, the weighted histogram H_w of a truecolor image I is defined as a total function:

$$H_w: \text{RGB} \rightarrow \mathbb{R} \quad (2.9)$$

where $H_w(c)$ is the total activity weighting of the pixels in I with color c . By simply substituting H_w for H , any histogram-based algorithm becomes activity weighted. Thus, a non-activity weighted, histogram-based algorithm may be viewed as an activity weighted, histogram-based algorithm in which all pixels are equally weighted as one. Sections 3.3, 0 and 5.2.2 show that activity weighting in this manner is a fast and effective way to enhance the quality of images produced by histogram-based color image quantization techniques.

2.3.2 Feedback-based Quantization

Xiang and Joy [XiJ94a and XiJ96] proposed a general feedback framework for improving the representative colors. The fundamental steps involved in their feedback-based quantization technique are delineated in Figure 2.3.

- 1) Quantize the original truecolor image I .
- 2) Detect visible distortion in the quantized image I_Q .
- 3) Use feedback to adjustment the weights of the pixels in I .
- 4) Repeat steps 1-3 until no further improvements can be made.

Figure 2.3: Pseudocode for Feedback-based Color Quantization.

Xiang and Joy [XiJ94a and XiJ96] use their agglomerative clustering technique [XiJ94b] in step 1 of Figure 2.3; however, any activity weighted technique may be used. Xiang and Joy [XiJ94a and XiJ96] detect visible distortion via a false contour map. The weight of a

pixel in I is then increased based on the severity of the false contour and the degree of quantization error. In practice, Xiang and Joy [XiJ94a and XiJ96] found no reliable method for automatically determining when no further improvements could be made. Thus, step 4 was implemented by running the algorithm for a block of iterations, viewing the sequence of quantized images, and resuming execution for another block of iterations based on the human observer's judgement. Because the feedback-based algorithm requires human interaction and large amounts of time (1-2 hours), this dissertation does not further investigate feedback-based quantization techniques.

2.3.3 Texture Analysis

Shufelt [Shu95 and Shu97] proposed a novel texture analysis technique for reducing false contours. His technique uses co-occurrence statistics of the quantized image to identify two classes of colors: contouring colors and compressible color pairs. Contouring colors are suspected of creating objectionable contouring artifacts in the quantized image, while it is believed that compressible color pairs may be merged without producing visible artifacts in the quantized image. The representative colors are then adjusted by splitting contouring colors and merging compressible color pairs. This process may be repeated until the texture analysis process fails to find any contouring colors or compressible color pairs.

The amount of correction required by the texture analysis is dependent upon the initial algorithm used to select the representatives. Shufelt uses Heckbert's median-cut algorithm with $p=(5,5,5)$ [Hec80]. It is well-known that many quantization techniques outperform the median-cut algorithm in terms of image quality (Section 3.3). Interestingly, Shufelt discovered that the improvement gained by texture analysis was comparable to the improvement gained by using full 24-bit precision. Given these observations, this dissertation does not further explore texture analysis, but rather focuses on developing methods for efficiently using full 24-bit precision.

2.4 Distortion Metrics

The goal of color image quantization is to *minimize the amount of distortion perceived* by a human observer between the truecolor image I and its color-mapped image I_Q . Unfortunately, the wondrous sophistication of the human visual system complicates the quantification of a perceptually-based distortion metric. Image compression and enhancement researchers have explored the subjective role of the observer in evaluating image quality and have proposed image metrics which attempt to quantify image quality [Sak77, Lim79, Gir83, Dal83, Lu91, Hul90, and GrW91]. However, the quantification of a perceptually-based image metric remains largely an unsolved problem. Thus researchers have been forced to concede and use distortion metrics which may not adequately capture the visual distortion [GoW92]. A few of the most common distortion metrics are described in the following sections.

2.4.1 RMSE

A common distortion metric used by image processing researchers is the root mean-square error, RMSE:

$$RMSE(I, I_Q) = \sqrt{\frac{1}{W \cdot H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} d(I(x, y), I_Q(x, y))^2} \quad (2.10)$$

where $d(I(x, y), I_Q(x, y))$ measures the distance between the color of the pixel at column x , row y of the truecolor image I and the color-mapped image I_Q . The most common distance function used is the Euclidean distance. The Euclidean distance between two colors $\mathbf{x}, \mathbf{y} \in \text{RGB}$ is

$$\|\mathbf{x} - \mathbf{y}\| = d_2(\mathbf{x}, \mathbf{y}) = d_{\text{RGB}}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}.R - \mathbf{y}.R)^2 + (\mathbf{x}.G - \mathbf{y}.G)^2 + (\mathbf{x}.B - \mathbf{y}.B)^2}. \quad (2.11)$$

In 1982 Garey, Johnson and Witsenhausen [GaJW82] showed that the selection of R which produces a quantized image that minimizes Equation 2.10 is NP-complete. Thus, it is unlikely that we will find an efficient algorithm to solve the color quantization problem. Hence, researchers have concentrated their efforts on developing heuristics for *approximating* the color quantizer that minimizes RMSE.

2.4.2 WRMSE

RMSE assumes that all pixels are equally important, whereas in activity weighting this assumption is no longer valid. Thus, the weighted RMSE (WRMSE) is a metric that weights the differences between the pixels in I and I_Q according the activity weighting of the pixels in I :

$$WRMSE(I, I_Q) = \sqrt{\frac{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} d(I(x, y), I_Q(x, y))^2 \cdot w(x, y)}{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} w(x, y)}} \quad (2.12)$$

Sections 3.4, 4.3.2, 5.1.2, and 5.2.2 show that WRMSE is much better than RMSE in measuring the visual distortion between I and I_Q .

2.4.3 SNR

The signal-to-noise ratio (SNR) is another common distortion metric used by image compression researchers:

$$SNR(I, I_Q) = 10 \cdot \log_{10} \left[\frac{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} I(x, y)^2}{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} d(I(x, y), I_Q(x, y))^2} \right] \quad (2.13)$$

Here, the truecolor image is considered to be the signal, and the distance between the truecolor image and the quantized image is considered to be noise. As with RMSE, SNR will be measured in both RGB and YIQ.

2.4.4 Maximum Diameter

One objective commonly used in cluster analysis is to minimize the maximum diameter of the individual clusters of a K -split S :

$$\maxDiam(S) = \max \{ diam(s_k) \mid s_k \in S \} \quad (2.14)$$

where

$$diam(s_k) = \max \{ d_2(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in s_k \}. \quad (2.15)$$

Section 4.2.3 shows that *maxDiam* is not a reasonable metric for color quantization because it does not correspond well to subjective evaluation.

2.4.5 Perceptually Uniform Color Spaces

The RGB color space is not ideal for color quantization because it is not perceptually uniform. That is, it is not possible to predict the perceptual closeness of two RGB colors from the RGB Euclidean distance between them. What is needed is a color space where the Euclidean distance between two colors is proportional to the perceptual distance between them. Two such color spaces called $L^*u^*v^*$ (CIELUV) and $L^*a^*b^*$ (CIELAB) were defined in 1976 by the Commission Internationale de l'Éclairage (CIE). Wysezecki and Stiles [WyS82] provide a detailed discussion of these two color spaces. Transformation from RGB into these color spaces is nonlinear and requires a cube root computation.

Gentile, Allebach and Walowit [GeAW90] found that using CIELUV provided no significant improvement over RGB for several adaptive quantization techniques. Shufelt [Shu95] studied alternative color spaces (HSV, HLS [FoVFHP93], and CIELAB) for the median-cut quantization technique (Section 3.2.1). For his 64 image test set, Shufelt concluded that RGB performed best most often. In addition, because of the time-consuming cube root function required to transform each pixel in the image from RGB to CIELAB, Shufelt's results indicate that the time required for quantization in CIELAB was about four to five times more than RGB. Connolly and Fliess [CoF97] describe several approximation techniques for the transformation from RGB to CIELAB. The fastest method proposed used a look up table, provided medium accuracy, and required up to 32KB of storage. Given these results, this dissertation focuses only on the RGB color space and explores other venues for producing high quality color-mapped images quickly.

Even so, Figure 2.4 illustrates how the quantization techniques described in this dissertation may be extended from the RGB color space to a perceptually uniform color space such as CIELUV or CIELAB.

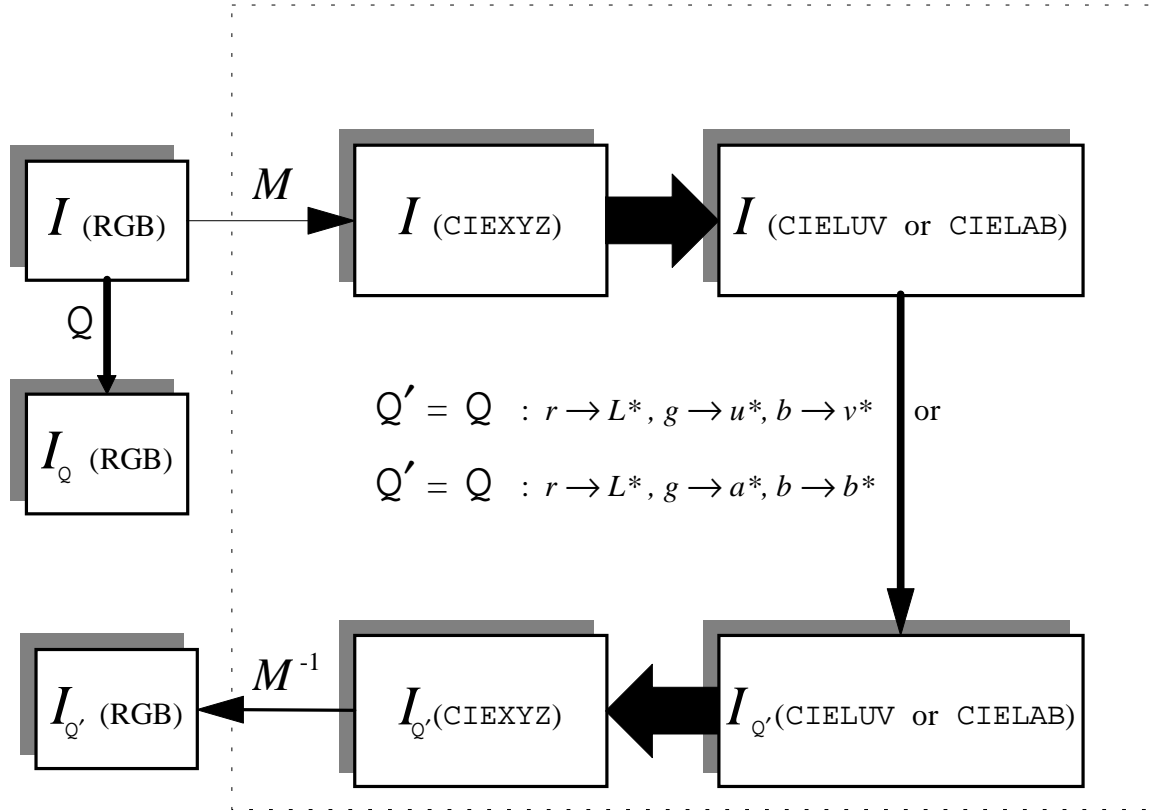


Figure 2.4: **Color Quantization in Perceptually Uniform Color Spaces.** M is a 3x3 transformation matrix determined by chromaticity data for the monitor's phosphors. The transformations between CIEXYZ, CIELUV and CIELAB are provided in [WyS82].

2.5 Strategy for Empirical Analysis

Ideally we are in search of both a color quantizer that minimizes perceived quantization artifacts and also an objective method for measuring these artifacts. It is not practical to physically examine and compare each color-mapped image to determine the magnitude and nature of visible distortion. Therefore, the objective distortion metrics described in Sections 2.4.1-2.4.4 are used to compare the fidelity of the color quantization techniques. RMSE and WRMSE are useful guides for comparing the quality of quantizers, but the human eye remains the final judge. Therefore, the author will subjectively evaluate a small subset of the quantized images produced by the empirical study. In addition, this

dissertation presents interesting examples of color-mapped images produced by the empirical study so that readers may observe the quality of the quantized images for themselves.

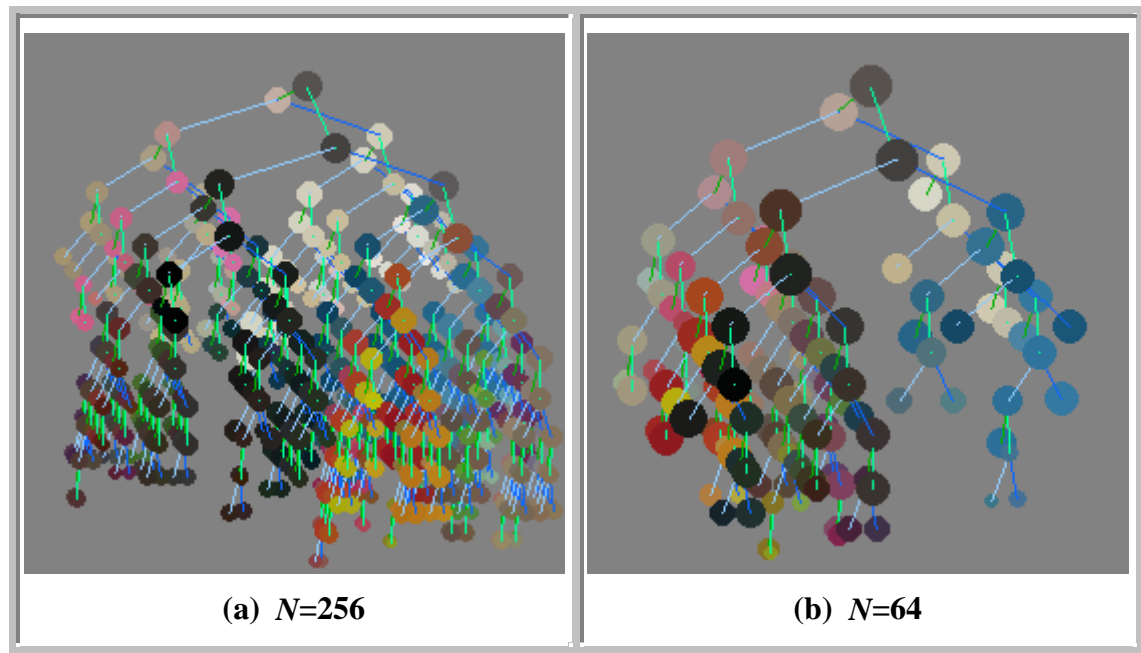
All of the color-mapped images were generated using LindyHop. LindyHop used the ANSI C `clock` function to capture the empirical timing data reported in this dissertation. LindyHop was compiled using `gnu g++` version 2.8.2 with optimization and was executed on a SUN Ultra 1 running Solaris 2.6 with 256 MB of RAM. Empirical timing data was captured when the machine was otherwise idle; that is, no other user processes were running. Since the measurements obtained from `clock` are subject to chance fluctuations, they will be treated as indicators of relative timing differences, not absolute truth.

This dissertation studies the interaction between K , p , time requirements, RMSE, WRMSE and subjective image quality. The primary goal of this dissertation is to study algorithms, techniques and parameters that produce high quality quantized images. Therefore, this dissertation primarily studies the effects of various algorithms and techniques for $K=256$ and $p=(8,8,8)$. However, empirical results for other values of K and p will be included for completeness and comparison. The analysis is simplified by not considering the potential speed-ups and the probable corresponding degradation in image quality resulting from subsampling (Section 2.2.3). The analysis is further simplified by not considering the potential improvement in image quality and the corresponding increase in quantization time resulting from dithering (Section 2.2.2).

Chapter 3

Hierarchical Methods

Hierarchical methods construct a tree depicting the relationship among the colors in the truecolor image I . Hierarchical clustering techniques that construct a tree from the bottom up (leaves to root) are called *agglomerative*, while hierarchical methods that construct a tree from the top down (root to leaves) are called *divisive*. Color Plate 3.1 illustrates two hierarchical clustering trees for (a) $N=256$ and (b) $N=64$ colors.



Color Plate 3.1: **Hierarchical Clustering Trees.** The size of each node is scaled by the frequency of the color. The number of leaves in the tree is (a) 256 and (b) 64.

The hierarchical clusterings are depicted by binary trees that have been drawn using the approach described in [Rei98]:

- Nodes on the same [odd/even] level are aligned in the $[x/z]$ direction.
- On [odd/even] levels, left subtrees are positioned to the [left/behind] their parent; right subtrees are positioned to the [right/in front].
- Parents on [odd/even] levels are centered in the $[z/x]$ direction between its two children.
- Odd level edges between [left/right] subtrees and their parents are [light blue/blue]; even level edges are [light green/green]. This use of color helps the user to orient the visualization in 3D space.
- Subtrees are drawn isomorphically and symmetrically.

Notice that the nodes are nested. That is, from the agglomerative point of view, once two nodes have been merged they are permanently joined into a new node that may be merged in the future. From the divisive point of view, once a group of nodes have been divided, they are permanently separated and may be treated independently for future subdivisions [And73].

The first two sections describe hierarchically agglomerative and divisive methods for color quantization respectively. The chapter concludes with a comprehensive empirical analysis of these algorithms.

3.1 Agglomerative

The fundamental steps in the hierarchically agglomerative method originally proposed by Ward [War63] are delineated in Figure 3.1. Recall that C is the set of N unique colors in a truecolor image I .

- 1) Find two “optimal” colors in C to merge, c_i and c_j .
- 2) Remove c_i and c_j from C .
- 3) Merge c_i and c_j into c_{ij} . Add c_{ij} to C .
- 4) Repeat steps 1-3 until there are K colors in C . C is the set of K representative colors for the truecolor image.

Figure 3.1: Pseudocode for Agglomerative Color Quantization.

This form of agglomerative clustering is commonly referred to as pairwise nearest neighbor (PNN) clustering because the cost of each pairwise merger must be evaluated to find the optimal pair of colors to merge in step 1 of Figure 3.1.

Equitz [Equ89] applied Ward’s general-purpose hierarchically agglomerative clustering algorithm to vector quantization and derived a specific formula for selecting the optimal colors to merge in step 1 of Figure 3.1. The optimal colors selected are two colors whose merger will result in the minimum increase in total squared error, TSE. Figure 3.1 shows a clever method for calculating the increase in total squared error (TSE_{\uparrow}) induced by merging two colors and is due to [Equ89]. Figure 3.1 has been generalized to enable the method of spatial activity weighting proposed in Section 2.3.1.

Theorem 3.1: $TSE_{\uparrow} = \frac{w_i w_j}{w_i + w_j} \|c_i - c_j\|^2$

Let H be a histogram for a truecolor image I in RGB space. Let c_i and c_j be two colors in H . Let $w_i = H(c_i)$ and $w_j = H(c_j)$ be the weight associated with c_i and c_j . If H is activity weighted, then w represents total activity weighting; otherwise, it represents frequency. Let TSE_i and TSE_j be the TSE associated with c_i and c_j respectively. Let c_{ij} be the cluster formed by merging c_i and c_j . The weight, value (mean), and total squared error of c_{ij} are defined as

$$w_{ij} = w_i + w_j \tag{1}$$

$$c_{ij} = \mu_{ij} = \frac{w_i c_i + w_j c_j}{w_{ij}}, \text{ and} \quad (2)$$

$$TSE_{ij} = TSE_i + TSE_j + \|c_i - \mu_{ij}\|^2 w_i + \|c_j - \mu_{ij}\|^2 w_j. \quad (4)$$

Then the increase in total squared error that results from merging c_i and c_j into c_{ij}

$$TSE_{\uparrow} = \frac{w_i w_j}{w_i + w_j} \|c_i - c_j\|^2.$$

Proof:

From Equation (4), the increase in TSE that results from merging c_i and c_j into c_{ij} is

$$\begin{aligned} TSE_{\uparrow} &= TSE_{ij} - (TSE_i + TSE_j) \\ &= \|c_i - \mu_{ij}\|^2 w_i + \|c_j - \mu_{ij}\|^2 w_j \end{aligned} \quad (5)$$

Using the definitions of w_{ij} and μ_{ij} , Equations (1-2), Equation (5) is expanded to

$$\begin{aligned} TSE_{\uparrow} &= \left\| c_i - \frac{w_i c_i + w_j c_j}{w_i + w_j} \right\|^2 w_i + \left\| c_j - \frac{w_i c_i + w_j c_j}{w_i + w_j} \right\|^2 w_j \\ &= \left\| \frac{w_i c_i + w_j c_i - w_i c_i - w_j c_j}{w_i + w_j} \right\|^2 w_i + \left\| \frac{w_i c_j + w_j c_j - w_i c_i - w_j c_j}{w_i + w_j} \right\|^2 w_j \\ &= \left\| \frac{w_j c_i - w_j c_j}{w_i + w_j} \right\|^2 w_i + \left\| \frac{w_i c_j - w_i c_i}{w_i + w_j} \right\|^2 w_j \\ &= \frac{w_j^2 w_i}{(w_i + w_j)^2} \|c_i - c_j\|^2 + \frac{w_i^2 w_j}{(w_i + w_j)^2} \|c_j - c_i\|^2 \\ &= \frac{w_j^2 w_i + w_i^2 w_j}{(w_i + w_j)^2} \|c_i - c_j\|^2 \\ &= \frac{w_i w_j (w_i + w_j)}{(w_i + w_j)^2} \|c_i - c_j\|^2 \\ &= \frac{w_i w_j}{(w_i + w_j)} \|c_i - c_j\|^2 \end{aligned} \quad \blacksquare$$

Thus, the only statistics required to calculate the increase in TSE induced by merging two colors are the value and weight of each color. Several researchers have studied

agglomerative color image quantization [Dix91, BaA91a, BaA91b, XiJ94b, XiJ94a, JoX96, PaLKLH96, VeGS97]. The method of activity weighting proposed by this dissertation differs from than the methods previously proposed by Balasubramanian et al. [BaA91a and BaA91b] :

$$TSE_{\uparrow} = \frac{n_i n_j}{n_i + n_j} \|c_i - c_j\|^2 \max(w_i, w_j), \quad (3.1)$$

and by Park et al. [PaLKLH96]:

$$TSE_{\uparrow} = \frac{n_i n_j}{n_i + n_j} \|c_i - c_j\|^2 \frac{w_i w_j}{w_i + w_j}, \quad (3.2)$$

where n_i and n_j represent frequency and w_i and w_j represent activity weighting. Thus, in both of the previous methods for activity weighted agglomerative cluster both frequency and activity weightings of the colors must be maintained. However, in the activity weighted method advocated by this dissertation, the frequency of the colors does not need to be maintained.

3.1.1 Fast PNN using k -d Trees

The primary drawback of agglomerative clustering is the time required by step 1 of Figure 3.1 to find the pair of colors to merge. Since the steps 2-4 of Figure 3.1 require constant time and are repeated $N-K$ times, they require $O(N-K)$ time. Unfortunately, the first step of Figure 3.1 requires that TSE_{\uparrow} be calculated $|C|^2/2$ times. Since $|C|$ is initially N and is decremented by one each time the loop is executed, the time required for the PNN algorithm is

$$\begin{aligned} T_{PNN}(N, K) &= O(N - K) + \frac{1}{2} \sum_{i=K+1}^N i^2 \\ &= O(N - K) + \frac{1}{2} \left(\sum_{i=1}^N i^2 - \sum_{i=1}^K i^2 \right) \\ &= O(N^3 - K^3). \end{aligned} \quad (3.3)$$

One of the main contributions of Equitz [Equ89] paper is the use of a k -d tree to speed the nearest neighbor searches in step 1 of Figure 3.1. The fundamental steps of the fast PNN algorithm proposed by Equitz are delineated in Figure 3.2.

- 1) Store the histogram of I in a k -d tree T . Let the maximum allowable bucket size be M . Let B be the number of buckets in the k -d tree.
- 2) For each bucket b_i , find two “optimal” colors in b_i to merge, c_{ii} and c_{ij} .
- 3) For a fixed fraction (γ) of the B buckets, remove c_{ii} and c_{ij} from b_i . Merge c_{ij} and c_{ij} into c_{iij} . Add c_{iij} to T .
- 4) If necessary, rebalance T . Split buckets containing more than M colors and merge small buckets.
- 5) Repeat steps 2-4 until there are K colors in T . T is the set of K representative colors for the truecolor image.

Figure 3.2: **Pseudocode for Fast PNN Color Quantization.**

The histogram of the truecolor image is first stored in a k -d tree whose maximum allowable bucket size is M . The nearest neighbor searches are sped up by limiting the search to pairs of colors in the same bucket of the k -d tree. The algorithm is further speed up by merging in each iteration a fixed fraction of the nearest neighbor pairs found in step 2 of Figure 3.2. The agglomerative color quantization algorithm proposed by Balasubramanian et al. [BaA91a and BaA91b] is actually an activity weighted version of Equitz’s fast PNN algorithm.

3.2 Divisive

Heckbert’s median cut algorithm [Hec80 and Hec82] was the first of a plethora of histogram-based, divisive algorithms proposed to select representative colors [WuW85, WaWP88, WaPW90, OrB91, Wu91b, Wu92, JoX93, RoG95, and BrB97]. The divisive algorithms statistically divide a pre-built histogram of I into K partitions; the centroid of each partition becomes one of the K representative colors. The fundamental steps involved in histogram-based divisive techniques are delineated in Figure 2.3.

- 1) Find the smallest RGB cube encompassing the colors in the truecolor image.
- 2) Repeatedly divide the RGB subpartition with the worst “measure” until you have divided the RGB cube into K subpartitions.
- 3) Select the centroids of the K subpartitions as the K representative colors for the truecolor image.

Figure 3.3: **Pseudocode for Histogram-Based Divisive Techniques.**

Pixel-based divisive techniques differ from the histogram-based divisive techniques by dividing the pixels in the image, rather than the colors in the image [ChTM94, KiLLH96a, and KiLLH96b]. Pixel-based divisive techniques enable each pixel to be weighted based on the context of its location in the image. This dissertation does not explore pixel-based divisive techniques; hence, from this point forward histogram-based divisive techniques will be referred to simply as divisive techniques.

Algorithms using the divisive approach must make several decisions. First a metric is needed to decide which partition (W) will be divided in step 2. Once W has been selected, the divisive approach must decide into how many partitions to divide W . Finally, the divisive approach must decide how to orient and where to position the partition plane(s).

Partition Selection	
N	Greatest number of <i>colors</i> .
P	Greatest number of <i>pixels</i> .
R	Greatest <i>range</i> in <i>R</i> , <i>G</i> or <i>B</i> .
σ^2	Greatest weighted <i>variance</i> .
σ_x^2	Greatest weighted <i>variance</i> in <i>R</i> , <i>G</i> or <i>B</i> .
E_{\downarrow}	Greatest <i>reduction</i> in error for a given partition plane orientation and position.
λ_1	Greatest <i>eigenvalue</i> of the covariance matrix.
Number of Subpartitions	
2	Split node into <i>two</i> partitions (<i>k</i> -d tree).
8	Split node into <i>eight</i> partitions using 3 orthogonal split planes (Octree).
N	Split node into $N > 2$ partitions using $N-1$ parallel split planes.
Partition Plane Orientation	
R	Orthogonal to the <i>R</i> , <i>G</i> or <i>B</i> axis with the greatest <i>range</i> .
σ^2	Orthogonal to the <i>R</i> , <i>G</i> or <i>B</i> axis with greatest weighted <i>variance</i>
E_{\downarrow}	Orthogonal to the <i>R</i> , <i>G</i> or <i>B</i> axis that maximizes the <i>reduction</i> in error for a given partition plane position.
μ_{3R}	Orthogonal to the line between μ_{3R} and μ .
e_1	Orthogonal to the first <i>principal component</i> of the covariance matrix.
Partition Plane Position	
C_D	<i>Center</i> of the domain.
C	<i>Center</i> of the range.
M	Weighted <i>median</i> .
μ	Weighted <i>mean</i> .
μ_{3R}	<i>Radius</i> weighted <i>mean</i> , one or (three)-dimensional.
t	<i>Optimal</i> position to maximize the reduction in projected (1D) error.
\perp	<i>Optimal</i> position to maximize the reduction in error for a family of parallel partition planes perpendicular to the <i>R</i> , <i>G</i> , or <i>B</i> axis.
\perp_e	<i>Optimal</i> position to maximize the reduction in error for a family of parallel partition planes perpendicular to an eigenvector.

Table 3.1: **Taxonomy of Statistics Used by Divisive Algorithms.**

Table 3.1 is a taxonomy of the various statistics used at the major decision points of the divisive approach. This taxonomy is a modification of a similar taxonomy developed by Sproull [Spr91] for classifying k -d tree construction methods. For example, using the proposed taxonomy, a divisive algorithm classified as **P2RM** is one that selects the partition with the largest number of *pixels*. The selected partition is divided into *two* subpartitions using a partition plane orthogonal to the R, G, or B axis with the largest *range*. Finally, the partition plane is positioned at the *median* of the R, G, or B axis with the largest range.

Thus, the divisive approach is a general framework for a large number of algorithms. Many of the possible divisive algorithms have been previously studied [Hec80, Hec82, WuW85, WaWP88, WaPW90, OrB91, Wu91b, Wu92, JoX93, ChTM94, RoG95, KiLLH96a, KiLLH96b and BrB97]. This dissertation proposes a 3D heterogeneous data structure to implement a hybrid divisive technique. Therefore, the remainder of this chapter is devoted to the description and analysis of specific divisive techniques and will make Table 3.1 more clear.

3.2.1 Median, Center, Mean and RWM-Cut

The algorithms discussed in this section position the partition plane orthogonal to one of the R, G, or B axes, and select the partition plane position based on the projected 1-dimensional distribution of the selected partition plane orientation. The light R, G, and B curves in Figure 3.5 show the projected R, G, and B distributions of *Windsails*. The median, center, mean and radius-weighted mean (RWM) of the R, G, and B distributions are indicated on the heavy horizontal R, G, and B lines in Figure 3.5. Let w_i be the weight level i occurs a 1-dimensional distribution with range $[l..u]$. The definition of weight has been generalized to accommodate the method of spatial activity weighting proposed in 2.3.1. That is, if the distribution is activity weighted, then w_i represents total activity weighting; otherwise, it represents frequency. The weight (W), median (m), center (c), mean (μ), and RWM (μ_R) are defined as follows:

$$W = \sum_{i=l}^u w_i, \quad (3.4)$$

$$m \text{ is such that } \sum_{i=l}^m w_i \cong \sum_{i=m+1}^u w_i \cong W / 2, \quad (3.5)$$

$$c = (l+u)/2, \quad (3.6)$$

$$\mu = \frac{\sum_{i=l}^u i w_i}{W}, \quad (3.7)$$

$$\mu_R = \frac{\sum_{i=l}^u |i - \mu| w_i}{\sum_{i=l}^u |i - \mu| w_i}. \quad (3.8)$$

The oft-cited *median-cut* algorithm was first described in Heckbert's B.S. thesis [Hec80] as the **R2RM** divisive algorithm. In Heckbert's later paper describing the median-cut algorithm [Hec82], the method for selecting a partition for subdivision is not made clear. The premise of the median-cut algorithm is to partition the histogram such that each partition represents approximately the same number of image pixels. Thus, the literature most commonly cites the median-cut algorithm as the **P2RM** divisive algorithm.

Wu and Witten [WuW85] described a divisive algorithm that is implemented slightly different from the divisive technique given in Figure 3.3. The fundamental steps in their approach are outlined in Figure 3.4.

- 1) Find the smallest RGB cube, W , encompassing the colors in the truecolor image.
- 2) Divide W into two subpartitions W_1 and W_2 . Split K into K_1 and K_2 based on a worst measure of the two subpartitions. K_1 and K_2 act as quotas for W_1 and W_2 .
- 3) Recursively divide (W_1, K_1) and (W_2, K_2) . Subdivision terminates when either K is one, or when the size of W is less than a predetermined minimal size. In the latter case, the excess quota is redistributed to remaining partitions.
- 4) Select the centroid of each of the K subpartitions as the K representative colors for the truecolor image.

Figure 3.4: **Pseudocode for Wu and Witten's Recursive Divisive Technique.**

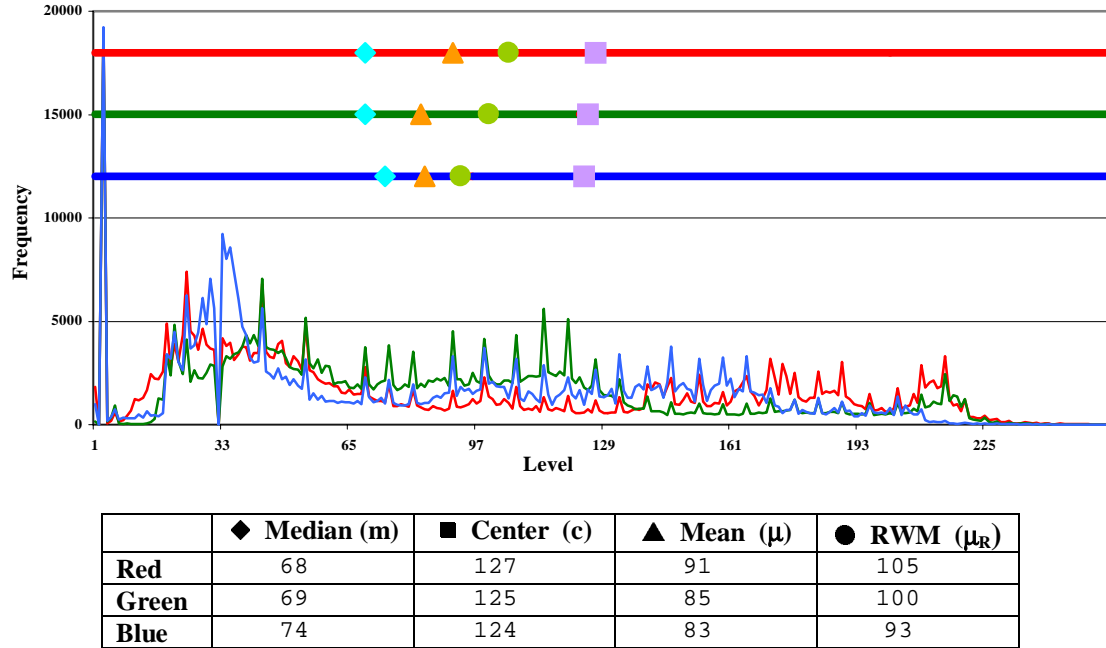


Figure 3.5: **Projected Distributions for *Windsails*, $p=(8,8,8)$.** The light R, G, and B curves show the projected R, G, and B distributions. The median, center, mean and radius-weighted mean of the R, G, and B distributions are indicated on the heavy horizontal R, G, and B lines.

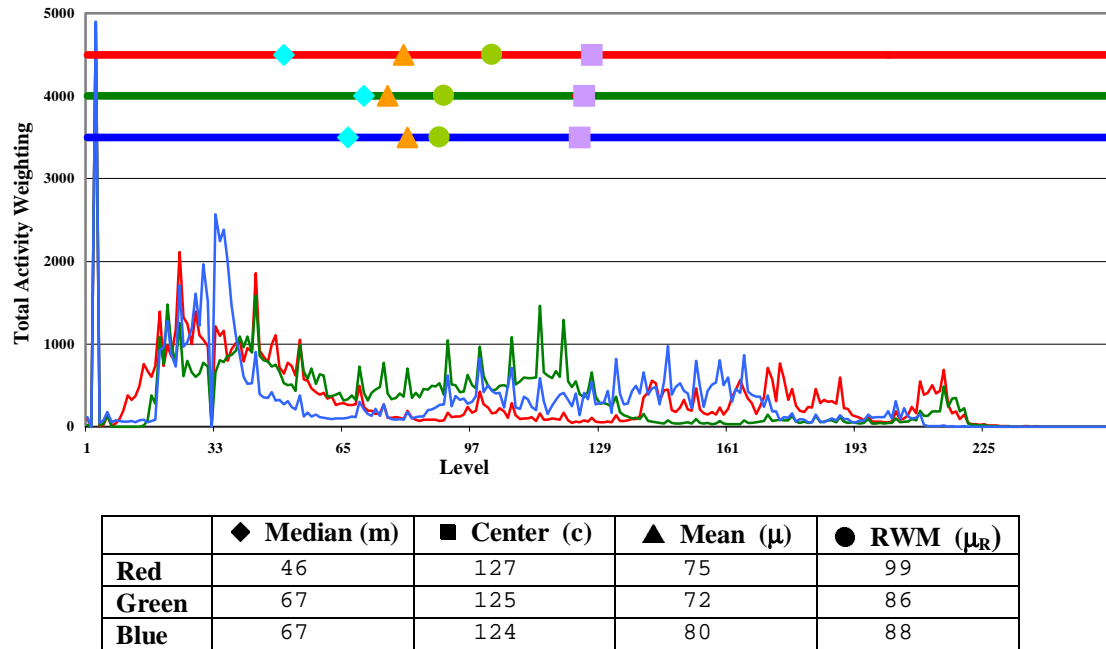


Figure 3.6: **Activity Weighted Projected Distributions for *Windsails*, $p=(8,8,8)$.** The curves are the same as Figure 3.5, except the distributions are activity weighted.

Chang and Chang [ChC93] proposed similar quota subdivision method that divided into eight instead of two pieces. Like the median-cut algorithm, Wu and Witten divide W into two subdivisions using a partition plane orthogonal to the R, G, or B axis with the greatest range. They allocate K based on the volume (V) and the cardinality (N) of the two subpartitions:

$$K_1 = \left[\rho \frac{N(C_1)}{N(C)} + (1 - \rho) \frac{V(C_1)}{V(C)} \right] \cdot K \text{ and } K_2 = K - K_1, \quad (3.9)$$

where $\rho \in [0,1]$, usually chosen in the range $[0.5, 0.7]$. Their algorithm is often cited as a *variance*-based algorithm because they claim Equation 3.9 approximates splitting K based on the variance of the two subpartitions. In this author's opinion, this claim is not fully justified, but illustrates the difference a decade has made in computer technology. Finally, Wu and Witten use a "center adjusted mean", rather than the actual mean, for positioning the partition plane. In particular, they use:

$$\mu_c = \mu + \eta(c - \mu), \quad (3.10)$$

where η is a constant in the range $[0.05, 0.25]$. Because $[0.05, 0.25]$ is a large range, this dissertation has opted not to explore the utility of a center-adjusted mean; that is, in this dissertation, $\eta=0$. For ease of coding and comparison with other divisive algorithms, this dissertation has implemented the spirit of Wu and Witten's *mean-split* algorithm [WuW85] using the divisive approach attributed to Heckbert [Hec80]. This dissertation will refer to the revised implementation of the mean-split algorithm as the *mean-cut* algorithm.

Joy and Xiang [JoX93] proposed a simple modification to Heckbert's original median-cut algorithm: split at the center instead of the median. Thus, this **R2RC** divisive algorithm is appropriately named the *center-cut* algorithm.

Yang and Lin [YaL96] proposed a variance-based algorithm that positions the partition plane at the RWM. Thus, this $\sigma^2 2\sigma^2 \mu_R$ divisive algorithm is appropriately named the *RWM-cut* algorithm.

3.2.2 Variance Minimization

Heckbert briefly describes a greedy *variance-minimization* technique in his B.S. thesis [Hec80]. His suggestion was to consider all possible split planes orthogonal to the dominant direction (assumed to be range) and calculate the sum of the variance of the two subpartitions induced. The point that induces the smallest sum is selected to be the split point. Heckbert did not implement his idea of greedy variance-minimization, however several variations on the theme have been studied [WaWP88, WaPW90, Wu91b, LiC94, LiC95, BrB97].

Liu and Chang [LiC95] proposed a greedy “lookahead” strategy for selecting the partition to be subdivided. Given that a partition will be divided into two subpartitions, and given a method for orienting and for positioning the cut plane, the partition selected for further subdivision is the one that will result in the most reduction in error. This strategy for selecting a partition is denoted as \mathbf{E}_\downarrow . Theorem 3.2 shows a clever method for calculating the reduction in error induced by a two-way split and is due to [LiC95]. Theorem 3.2 has been generalized to enable the method of spatial activity weighting proposed in Section 2.3.1.

$$\text{Theorem 3.2: } TSE_\downarrow = W_{l_1} \|\mu_{l_1} - \mu_l\|^2 + W_{l_2} \|\mu_{l_2} - \mu_l\|^2$$

Let S be a k -d tree containing a histogram, H , of a truecolor image, I , in RGB space. Let $w_c = H(c)$ be the weight associated with c . If H is activity weighted, then w_c represents total activity weighting; otherwise, it represents frequency. Let l be a leaf in S . The weight, mean, and total squared error of l are defined as

$$W_l = \sum_{c \in l} w_c, \tag{1}$$

$$\mu_l = \frac{\sum_{c \in l} cw_c}{W_l}, \text{ and} \tag{2}$$

$$TSE_l = \sum_{c \in l} \|c - \mu_l\|^2 w_c, \tag{4}$$

where $n_c = H(c)$ is the number of times the color c occurs in I . Then the reduction in total squared error that results from dividing l into two leaves l_1 and l_2 is

$$TSE_{\downarrow} = W_{l_1} \|\mu_{l_1} - \mu_l\|^2 + W_{l_2} \|\mu_{l_2} - \mu_l\|^2.$$

Proof:

Let L be the set of leaves in S . Then the TSE in S before splitting l is

$$TSE = \sum_{i \in L} TSE_i = \sum_{\substack{i \in L \\ i \notin l}} TSE_i + TSE_l. \quad (4)$$

Clearly, the TSE in S after splitting l into two leaves l_1 and l_2 is

$$TSE' = \sum_{\substack{i \in L \\ i \notin l}} TSE_i + TSE_{l_1} + TSE_{l_2}. \quad (5)$$

Therefore, from Equations (4) and (5), the reduction in TSE that results from dividing l into two leaves l_1 and l_2 is

$$TSE_{\downarrow} = TSE - TSE' = TSE_l - TSE_{l_1} - TSE_{l_2}. \quad (6)$$

Since l is split into two leaves, l_1 and l_2 , the TSE at l can be computed as

$$\begin{aligned} TSE_l &= \sum_{c \in l_1} \|c - \mu_l\|^2 n_c + \sum_{c \in l_2} \|c - \mu_l\|^2 w_c \\ &= \sum_{c \in l_1} \|(c - \mu_{l_1}) + (\mu_{l_1} - \mu_l)\|^2 w_c + \sum_{c \in l_2} \|(c - \mu_{l_2}) + (\mu_{l_2} - \mu_l)\|^2 w_c. \end{aligned} \quad (7)$$

Using the identity, $\|x+y\|^2 = \|x\|^2 + 2 \cdot xy + \|y\|^2$, Equation (7) is expanded to:

$$\begin{aligned} TSE_l &= \sum_{c \in l_1} \left(\|c - \mu_{l_1}\|^2 + 2(c - \mu_{l_1})(\mu_{l_1} - \mu_l) + \|\mu_{l_1} - \mu_l\|^2 \right) w_c + \\ &\quad \sum_{c \in l_2} \left(\|c - \mu_{l_2}\|^2 + 2(c - \mu_{l_2})(\mu_{l_2} - \mu_l) + \|\mu_{l_2} - \mu_l\|^2 \right) w_c \end{aligned} \quad (8)$$

Using the definitions of W , μ , and TSE of a leaf, Equations (1-3), Equation (8) is reduced to

$$\begin{aligned}
TSE_l &= TSE_{l_1} + TSE_{l_2} + \\
&\quad 2(\mu_{l_1} - \mu_l) \sum_{c \in l_1} (c - \mu_{l_1}) w_c + 2(\mu_{l_2} - \mu_l) \sum_{c \in l_2} (c - \mu_{l_2}) w_c + \\
&\quad W_{l_1} \|\mu_{l_1} - \mu_l\|^2 + W_{l_2} \|\mu_{l_2} - \mu_l\|^2 \\
&= TSE_{l_1} + TSE_{l_2} + W_{l_1} \|\mu_{l_1} - \mu_l\| + W_{l_2} \|\mu_{l_2} - \mu_l\| + \\
&\quad 2(\mu_{l_1} - \mu_l) \left[\sum_{c \in l_1} c w_c - \mu_{l_1} W_1 \right] + 2(\mu_{l_2} - \mu_l) \left[\sum_{c \in l_2} c w_c - \mu_{l_2} W_2 \right] \\
&= TSE_{l_1} + TSE_{l_2} + W_{l_1} \cdot \|\mu_{l_1} - \mu_l\|^2 + W_{l_2} \cdot \|\mu_{l_2} - \mu_l\|^2
\end{aligned} \tag{9}$$

Hence, from Equations (6) and (9)

$$TSE_{\downarrow} = W_{l_1} \|\mu_{l_1} - \mu_l\|^2 + W_{l_2} \|\mu_{l_2} - \mu_l\|^2 .$$

■

Liu and Chang select the mean as the partition position and then perform an additional level of greedy lookahead to decide which of the R, G, or B axis to cut with an orthogonal plane. Not surprisingly, they choose the direction that maximizes the reduction in error. Thus, the greedy algorithm proposed by Liu and Chang [LiC95] uses two levels of lookahead and Theorem 3.2 to derive the **E_↓2E_↓μ** divisive algorithm.

The variance-minimization algorithm proposed by Wan, Wong, and Prusinkiewicz [WaWP88 and WaPW90] always divides the partition with the largest variance with an axis parallel split plane with the goal of maximizing the reduction in variance. Instead of considering all possible cut planes perpendicular to the R, G, and B axes, they used the projected distributions to calculate the “optimal” 1-dimensional cutpoints in each of the R, G, and B directions. Using Equations (3.4) and (3.7) the optimal (*t*) cutpoint for a 1-dimensional distribution with range [*l..u*] is defined as:

$$t = \arg \max_{\frac{l+\mu}{2} \leq t \leq \frac{\mu+u}{2}} E_{\downarrow}(t), \text{ where} \quad (3.11)$$

$$E_{\downarrow}(t) = \left[\frac{W_1}{W_2} (\mu - \mu_1)^2 \right], \quad (3.12)$$

$$W_1 = \sum_{i=l}^t w_i, \quad (3.13)$$

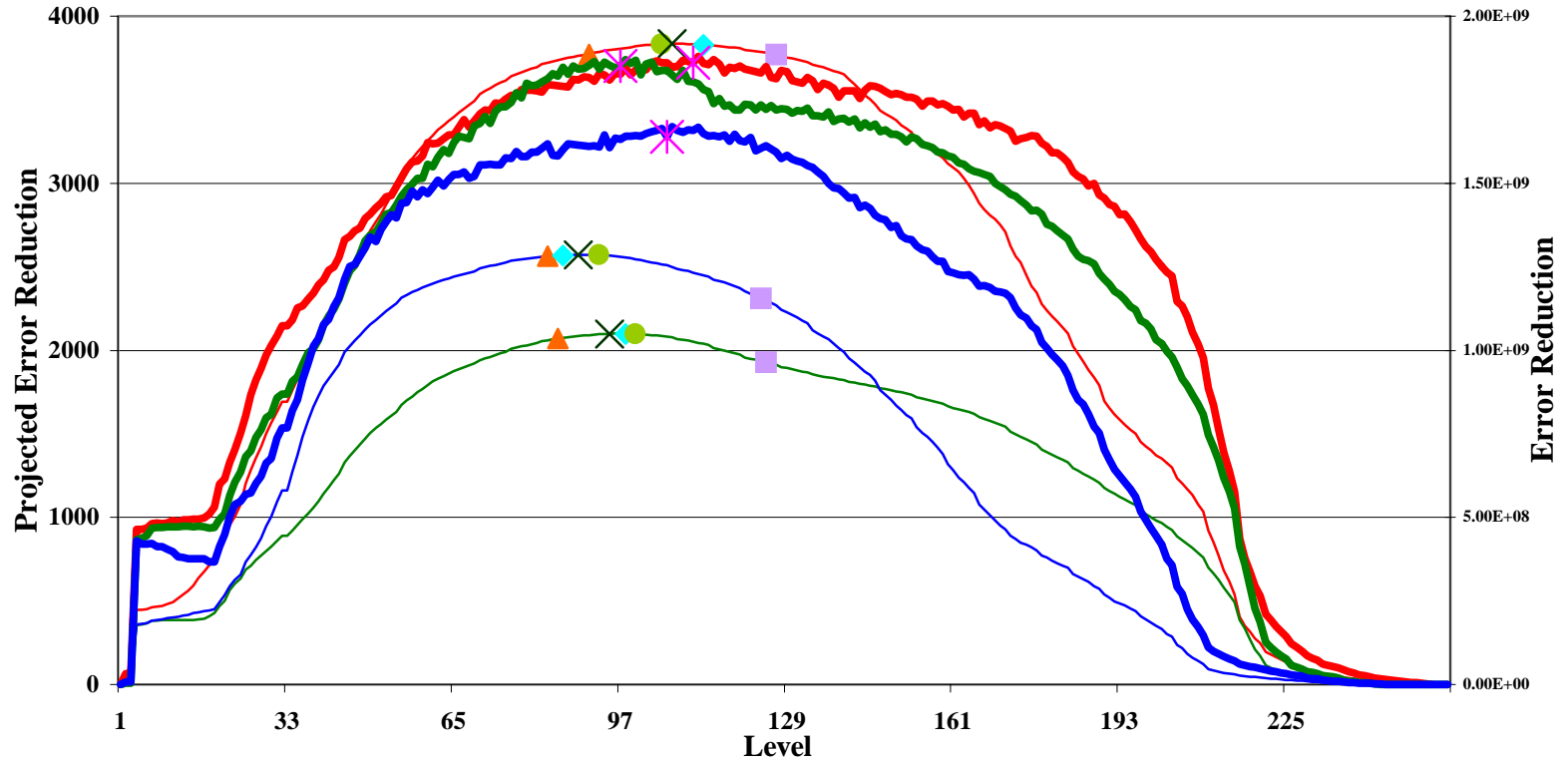
$$W_2 = \sum_{i=t+1}^u w_i, \text{ and} \quad (3.14)$$

$$\mu_1 = \frac{\sum_{i=l}^t i w_i}{W_1}. \quad (3.15)$$

As with Liu and Chang, Wan et al. split in the direction that maximizes the reduction in variance, resulting in the $\sigma^2 2E_{\downarrow} \perp$ divisive algorithm.

Several researchers have studied divisive algorithms which calculate the optimal cutpoints as defined by Heckbert; that is, for a family of parallel split planes, the one that maximizes TSE_{\downarrow} is chosen. This strategy for selecting a partition plane position is denoted as \perp . Wu [Wu91b] proposed the $\sigma^2 2E_{\downarrow} \perp$ algorithm. Later Braquelaire and Brun [BrB97] studied a small variation on Wu's algorithm, $\sigma^2 2\sigma^2 \perp$.

For example, the light R, G, and B curves in Figure 3.7 show E_{\downarrow} for each of the R, G, and B projected distributions of *Windsails* (left y-axis). The heavy R, G, B curves show TSE_{\downarrow} induced by sweeping an orthogonal cut plane along each of the R, G, and B directions (right y-axis). The median, center, mean, RWM and optimal cut point of the R, G, and B distributions are indicated on the E_{\downarrow} curves. The optimal 1-dimensional cutpoints defined by Wan et al. are the levels at which the E_{\downarrow} curves obtain maximal value (**X**). The optimal cutpoints defined by Heckbert are the levels at which the TSE_{\downarrow} curves obtain maximal value (*****). Note that the projected distributions indicate that red is the best axis to split and that blue is a distant second choice, but actually both red and green are good choices.



	◆ m	■ c	▲ μ	● μ _R	X t	$E_{\downarrow}(t)$	$TSE_{\downarrow}(t)$	* ⊥	$TSE_{\downarrow}(\perp)$	TSE
Red	113	127	91	105	107	3834.38	1.85E+09	111	1.88E+09	1.83E+09
Green	98	125	85	100	95	2099.68	1.85E+09	97	1.87E+09	1.22E+09
Blue	86	124	83	93	89	2571.91	1.61E+09	106	1.67E+09	1.29E+09

Figure 3.7: **Error Reduction for Windsails, $p=(8,8,8)$.** The light R, G, B curves show E_{\downarrow} for each of the projected R, G, and B distributions of Windsails. The heavy R, G, B curves show TSE_{\downarrow} induced by sweeping an orthogonal cut plane along each of the R, G, and B directions (right y-axis).

Recently Kim et al. [KiLLH96a and KiLLH96b] proposed selecting the partition plane position using the “optimal” grayscale threshold method developed by Otsu [Ots79]. This author has verified that the optimal greyscale threshold defined by Otsu and the optimal cutpoint defined by Wan et al. [WaWP88] are equivalent; they are simply defined using different definitions and notations.

3.2.3 Oct-cut

All of the algorithms discussed so far have split the “worst” partition into two pieces. The oct-cut divisive algorithms split the “worst” partition into eight pieces, whereby constructing an octree vice a k -d tree. The oct-cut method, σ^28t , was first introduced in connection with pixel-based activity weighted techniques [ChTM94, KiLLH96a, and KiLLH96b] where higher quality images were sought by incorporating spatial information. A very simple oct-cut method, $P8C_D$, was proposed by Roytman and Gotsman [RoG95] to dynamically quantize video sequences. In this case, speed was the primary consideration.

In previous work, when the number of remaining colors to be chosen was less than seven, the subdivision process terminated. Hence, up to six representative colors may be unused. This dissertation proposes a small variation on the oct-cut method to prevent under-utilizing representative colors: when the number of colors to be selected is less than seven, subdivide the partition into two pieces instead of eight. Thus, the data structure built is a heterogeneous tree consisting of octree and k -d tree nodes.

3.2.4 Oblique-Cut

All of the algorithms discussed so far have restricted the partition planes to axis parallel cuts. This section looks at two methods that use partition planes that are oblique to the R, G, or B axis. The first method uses principal component analysis [OrB91, Wu91a, WuZ91, BaBA92 and Wu92], and the second method uses the 3D radius weighted mean [YaL95 and YaL96].

Principal Component Analysis

Principal component analysis is used to split the partition orthogonal to the direction of greatest variance, which may not coincide with one of the R, G, or B axis. Principal component analysis requires the covariance matrix of each subpartition be maintained. Let S be a subset of a (possibly activity weighted) histogram, H , for a truecolor image I . Let $w_c = H(c)$ be the weight of c . The covariance matrix $CoVar$ of S is defined as

$$CoVar(S) = \sum_{c \in S} c c^t w_c - \frac{\left(\sum_{c \in S} c w_c \right) \left(\sum_{c \in S} c w_c \right)^t}{\sum_{c \in S} w_c}. \quad (3.16)$$

The covariance matrix is a symmetric 3x3 positive semidefinite matrix. For example, the covariance matrix for *Windsails* is

$$CoVar(H_{Windsails}) = \begin{bmatrix} 1.826\text{E} + 09 & 9.6923\text{E} + 08 & 5.386\text{E} + 08 \\ 9.6923\text{E} + 08 & 1.216\text{E} + 09 & 9.988\text{E} + 08 \\ 5.386\text{E} + 08 & 9.988\text{E} + 08 & 1.287\text{E} + 09 \end{bmatrix}.$$

The diagonal entries of the covariance matrix are the variances in the R, G, and B directions. The diagonal entries for *Windsails* covariance matrix are equivalent to the TSEs reported in Figure 3.7.

The trace of the covariance matrix is the total variance of S . Therefore, the total variance for *Windsails* is 4.32908e+09. The off diagonal entries are the covariances of two unique (R, G, and B) directions. For instance, the covariance of red and green for *Windsails* is 969236586.392. If the covariance of two unique directions is zero, then the two directions are uncorrelated.

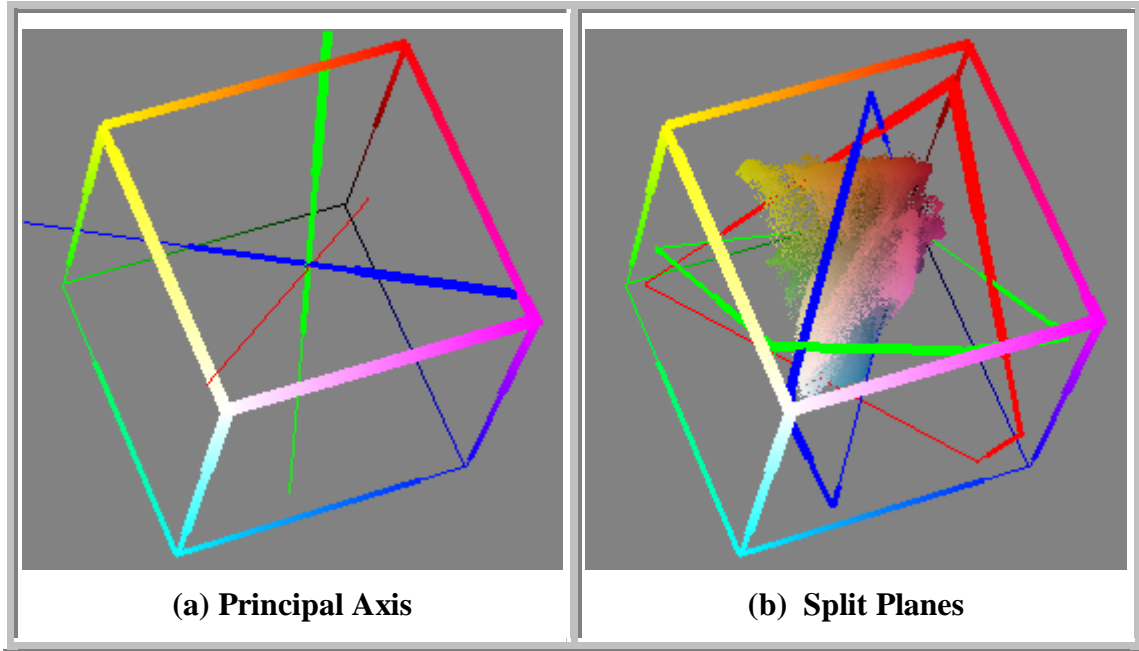
The eigenvector transform of the covariance matrix is an orthogonal matrix with the property that the transformed data is uncorrelated. The unit eigenvectors of the covariance matrix are called the principal components of S . The eigenvalues are ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The first principal component, \mathbf{e}_1 , is the eigenvector corresponding to the largest eigenvalue, λ_1 . Therefore, λ_1 represents the amount of variance in the direction with the greatest variance, \mathbf{e}_1 . The eigenvalues and corresponding principal components of *Windsails* are

$$\begin{array}{ccc}
\lambda_1=3.135\text{E}+09 & \lambda_2=1.035\text{E}+09 & \lambda_3=1.592\text{E}+08 \\
\mathbf{e}_1 = \begin{bmatrix} 0.639 \\ 0.584 \\ 0.502 \end{bmatrix} & \mathbf{e}_2 = \begin{bmatrix} 0.720 \\ -0.222 \\ -0.658 \end{bmatrix} & \mathbf{e}_3 = \begin{bmatrix} 0.273 \\ -0.781 \\ 0.561 \end{bmatrix}
\end{array}$$

The total variance of S is $\lambda_1 + \lambda_2 + \lambda_3 = \text{tr}(\text{CoVar}(S))$. The quotient $\lambda_i/\text{tr}(\text{CoVar}(S))$ is the fraction of variance in direction of the i^{th} principal component. In the case of *Windsails*, the percentages of variance in the directions of the principal components are

$$\begin{array}{ccc}
\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\
\frac{3.135\text{E} + 09}{4.329\text{E} + 09} \cdot 100 = 72.4\% & \frac{1.035\text{E} + 09}{4.329\text{E} + 09} \cdot 100 = 23.9\% & \frac{1.592\text{E} + 08}{4.329\text{E} + 09} \cdot 100 = 3.7\%
\end{array}$$

Color Plate 3.2 illustrates the principal components of *Windsails*. Orchard and Bouman [OrB91] split partition with the largest eigenvalue with a partition plane orthogonal to its first principal component and intersecting the mean, resulting in the $\lambda_1 \mathbf{2e}_1 \mu$ divisive algorithm. Yang and Lin [YaL94] proposed intersecting at the 3D RWM instead of the mean, resulting in the $\sigma^2 \mathbf{2e}_1 \mu_{3R}$. The 3D RWM is defined in the following section. Wu proposed [Wu91a] sweeping a cut plane perpendicular to the principal axis and selecting the position that maximizes the reduction in variance, resulting in the $\sigma^2 \mathbf{2e}_1 \perp_e$ divisive algorithm. By sorting the colors by their projections onto the principal axis, Theorem 3.2 may be used to calculate the reduction of variance induced by a split orthogonal to the principal axis. Wu also proposed [Wu92] using dynamic programming to find the globally optimal κ -partition ($\kappa > 2$) of the color space in the principal direction of the histogram. Typically, $4 \leq \kappa \leq 8$. Thus, if κ is less than K , the remaining $K - \kappa$ are chosen using the $\sigma^2 \mathbf{2e}_1 \perp_e$ divisive algorithm.



Color Plate 3.2: **Principal Components for Windsails.** (a) The first, second and third principal components are drawn in red, green and blue respectively. (b) The red, green, and blue split planes are orthogonal to the first, second and third principal components. The principal axis in (a) and split planes in (b) intersect at the mean (91,85,83).

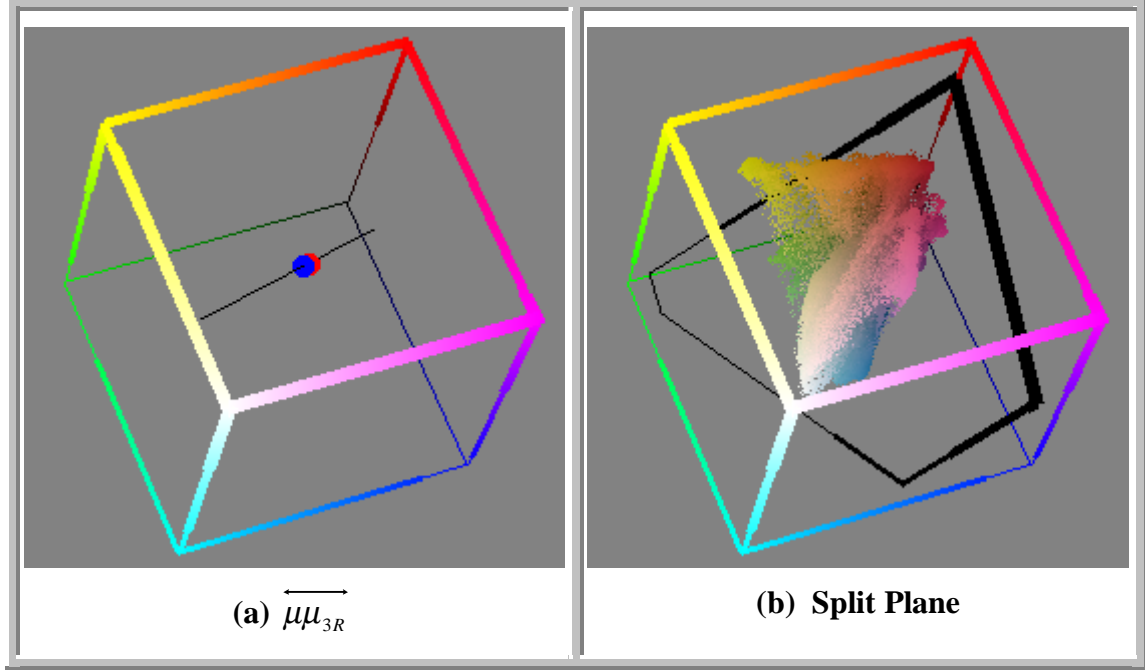
3D RWM

Yang and Lin proposed a novel divisive method that uses the 3D radius-weighted mean (RWM) [YaL95 and YaL96]. The 3D RWM of a set S (see previous section) is defined as

$$\mu_{3R} = \frac{\sum_{c \in S} c |c - \mu| w_c}{\sum_{c \in S} |c - \mu| w_c}. \quad (3.17)$$

Thus, the 3D RWM biases the mean value toward distant colors. Yang and Lin split the partition with the largest variance with a partition plane intersecting μ_{3R} and orthogonal to the line $\overrightarrow{\mu\mu_{3R}}$, resulting in the $\sigma^2_{\mu_{3R}}$ divisive algorithm. Color Plate 3.3 illustrates using the 3D RWM to obtain an oblique cut for *Windsails*. Notice how

remarkably close $\overrightarrow{\mu\mu_{3R}}$ is to the first principal axis of *Windsails* drawn in Color Plate 3.2(a).



Color Plate 3.3: **3D RWM for Windsails.** (a) The line passing through the μ (red) and μ_{3R} (blue) is drawn. (b) The split plane passes through the μ_{3R} (104,96,90) and is orthogonal to the line $\overrightarrow{\mu\mu_{3R}}$.

3.3 Empirical Analysis

This section provides an empirical analysis of the algorithms discussed in this chapter. The purpose of including the agglomerative algorithms in the study is to determine the relative quality merits between agglomerative and divisive hierarchical clustering. Therefore, the agglomerative algorithms studied do not incorporate any fast PNN techniques discussed in Section 3.1.1.

Table 3.2 lists the divisive algorithms studied in this section. Note that the mean-cut algorithm that corresponds closest to Wu and Witten's mean-split algorithm $\sigma^2 2R\mu$ [WuW85] was not included in the analysis. Figure 3.8 compares the average running

time and RMSE of all the algorithms listed in Table 3.2 for $K=256$ and $p=(8,8,8)$. This is the most comprehensive study of divisive algorithms ever conducted. For the first time, one can easily compare the performance of divisive algorithms.

As expected, the oct-cut algorithms are the fastest, but produce quantized images with relatively large RMSE. The non variance-based median-, center-, and mean-cut algorithms also produce quantized images with large RMSE. The variance-based median-, center-, and mean-cut algorithms, as well as the variance-minimization and the principal-component oblique-cut methods all produce quantized images with relatively low RMSE. Interestingly, the 3D RWM oblique-cut algorithms, $\sigma^2\mu_{3R}\mu_{3R}$ and $\sigma^2\mu_{3R}\mu_{3R}$, did not fair so well. They were faster than the principal component oblique-cut methods, but their RMSE was not as good as the better divisive algorithms.

As previously stated in Section 3.1.1, the primary drawback of PNN is the $O(N^3)$ time required to select the representative colors. Table 3.3 provides the running time, RMSE, and WRMSE for the PNN algorithm on the entire test set with $K=256$ and $p=(8,8,8)$. This table shows that the fastest running time (309s) is slow enough, but that on average, the running time is 64,039s which is approximately 17 hours and 47 minutes. *Eye* contains the largest number of unique colors and requires the most amount time: 4 days, 10 hours, and 45 minutes! The average RMSE for PNN is 5.43, but this is only 3.7% better than the best divisive algorithm analyzed in this section. This small decrease in average RMSE requires on average 3,241 times more time.

Next the effect activity weighting has on quantization time and image quality is examined. Figure 3.9 compares the average running time and WRMSE of all the activity weighted versions of the algorithms listed in Table 3.2. Comparing Figure 3.8 to Figure 3.9 shows that activity weighting does not significantly change the average running time of the algorithms. Also, the relative error produced by the algorithms remains largely the same. That is, the WRMSE produced by activity weighting is largely determined by the RMSE produced without activity weighting.

<i>Symbol</i>	<i>Section</i>	<i>Algorithm</i>	<i>Number</i>	<i>Classification</i>
♦	3.2.1	Median-Cut	50	R2RM
			51	$P2\sigma^2M$
			52	$N2\sigma^2M$
			53	$\sigma_x^2 2\sigma^2M$
			54	$\sigma^2 2\sigma^2M$
♦	3.2.1	Center-Cut	60	R2RC
			61	$P2\sigma^2C$
			62	$N2\sigma^2C$
			63	$\sigma_x^2 2\sigma^2C$
			64	$\sigma^2 2\sigma^2C$
♦	3.2.1	Mean-Cut	70	R2R μ
			71	$P2\sigma^2\mu$
			72	$N2\sigma^2\mu$
			73	$\sigma_x^2 2\sigma^2\mu$
			74	$\sigma^2 2\sigma^2\mu$
♦	3.2.1	RWM-Cut	80	$\sigma^2 2\sigma^2\mu_R$
♦	3.2.2	Variance-based	81	$\sigma^2 2\sigma^2t$
+	3.2.2	Variance-Minimization	90	$\sigma^2 2E_{\downarrow}\mu$
			91	$\sigma^2 2E_{\downarrow}t$
			92	$\sigma^2 2E_{\downarrow}\mu_R$
			94	$E_{\downarrow} 2E_{\downarrow}\mu$
			95	$E_{\downarrow} 2E_{\downarrow}t$
			96	$E_{\downarrow} 2E_{\downarrow}\mu_R$
			97	$\sigma^2 2\sigma^2\perp$
			98	$\sigma^2 2E_{\downarrow}\perp$
			99	$E_{\downarrow} 2E_{\downarrow}\perp$
–	3.2.3	Oct-Cut	100	P8C _D
			103	$\sigma^2 8\mu$
			104	$\sigma^2 8t$
			106	$\sigma^2 8\mu_R$
			107	$\sigma^2 8\perp$
•	3.2.3	3D RWM	120	$\sigma^2 2\mu_{3R}\mu_{3R}$
			121	$\sigma^2 2\mu_{3R}\mu$
•	3.2.3	PCA	130	$\sigma^2 2e_i\mu$
			131	$\lambda_1 2e_i\mu$
			132	$\sigma^2 2e_i\mu_{3R}$
			133	$\sigma^2 2e_i t$
			135	$\sigma^2 2e_i\perp_e$

Table 3.2: **List of Divisive Algorithms Analyzed.** The algorithms with red numbers are studied in greater detail than the other algorithms. The algorithms with pink classifications are being studied for the first time by this dissertation.

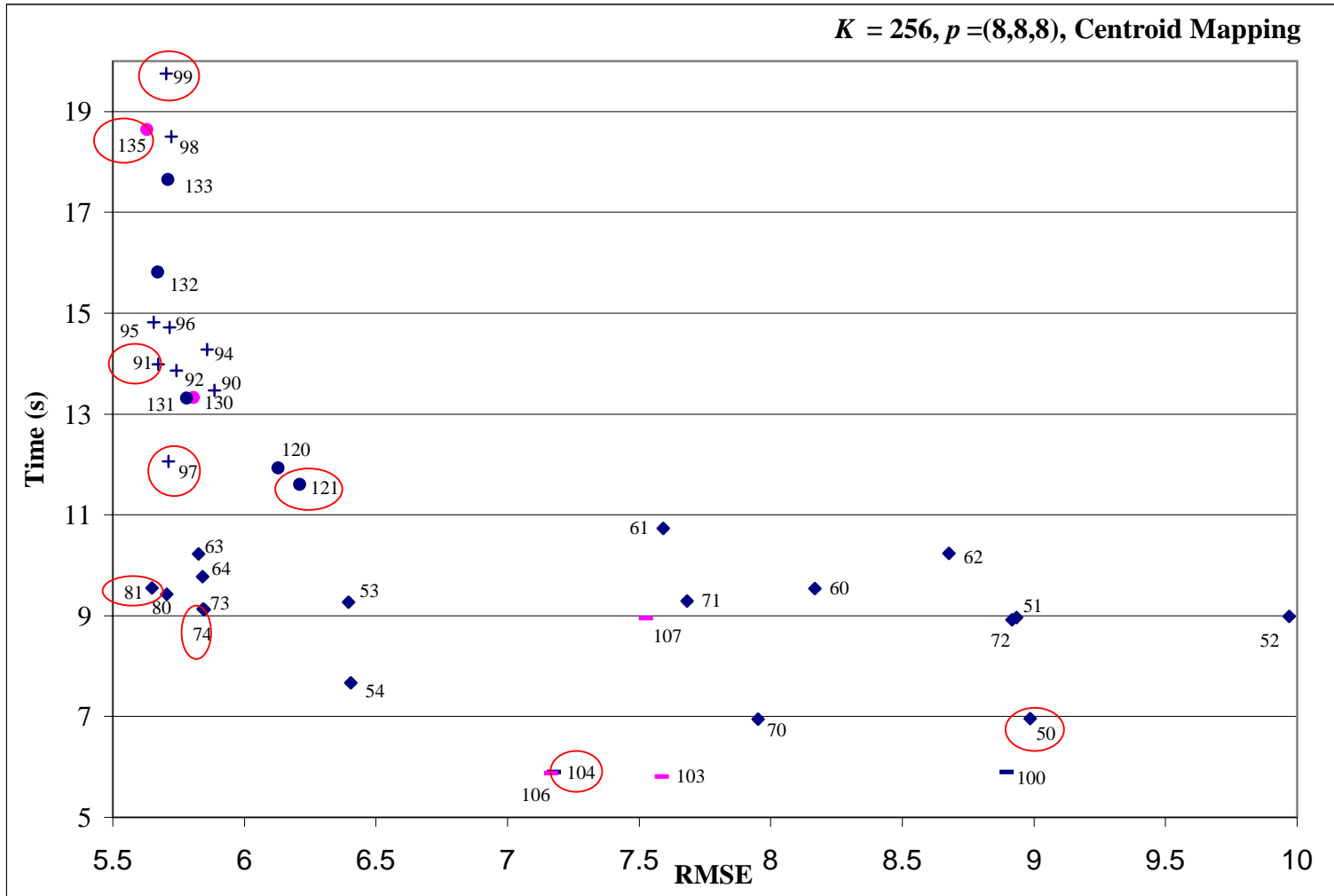


Figure 3.8: **Average Running Time and RMSE of Divisive Algorithms.** $K=256, p=(8,8,8)$.

#	Name	N	Time (s)	RMSE	WRMSE
1	<i>Shufelt</i>	20,302	1,783	4.37	2.74
2	<i>RGBCube</i>	34,111	5,078	8.10	8.87
3	<i>Shapes</i>	9,554	309	4.24	3.80
4	<i>Crambin</i>	19,073	1,634	5.30	4.78
5	<i>Solids</i>	31,429	4,677	3.53	2.88
6	<i>Marble</i>	21,532	2,332	4.14	3.76
7	<i>Cathedral</i>	28,662	4,239	5.08	4.19
8	<i>Sunset</i>	19,306	1,791	3.64	3.57
9	<i>Empire</i>	57,388	16,746	4.39	3.09
10	<i>Jhonni</i>	67,085	23,603	5.59	5.08
11	<i>Parrot</i>	40,565	8,455	4.17	3.46
12	<i>Matches</i>	64,112	21,301	4.86	4.70
13	<i>Mom</i>	96,486	46,185	7.55	6.30
14	<i>Tiger</i>	73,939	27,816	5.78	4.70
15	<i>Flowers</i>	71,687	27,254	6.20	5.77
16	<i>Peppers</i>	111,344	60,401	7.57	6.93
17	<i>Boys</i>	23,676	3,391	2.20	2.10
18	<i>Windsails</i>	86,008	38,277	6.93	5.70
19	<i>Cat</i>	95,916	50,034	3.93	3.47
20	<i>Woman</i>	94,279	45,568	7.22	6.11
21	<i>Lena</i>	148,279	119,934	5.90	5.51
22	<i>Boy</i>	180,274	181,874	5.11	4.64
23	<i>Eye</i>	267,246	384,340	6.52	6.23
24	<i>Chapel</i>	266,764	348,918	8.97	7.32
25	<i>Sailing</i>	177,844	175,022	4.56	4.02
	Max	267,246	384,340	8.97	8.87
	Min	9,554	309	2.20	2.10
	Avg	84,274	64,039	5.43	4.79

Table 3.3: **Running Time, RMSE, and WRMSE for PNN.** $K=256$, $p=(8,8,8)$.

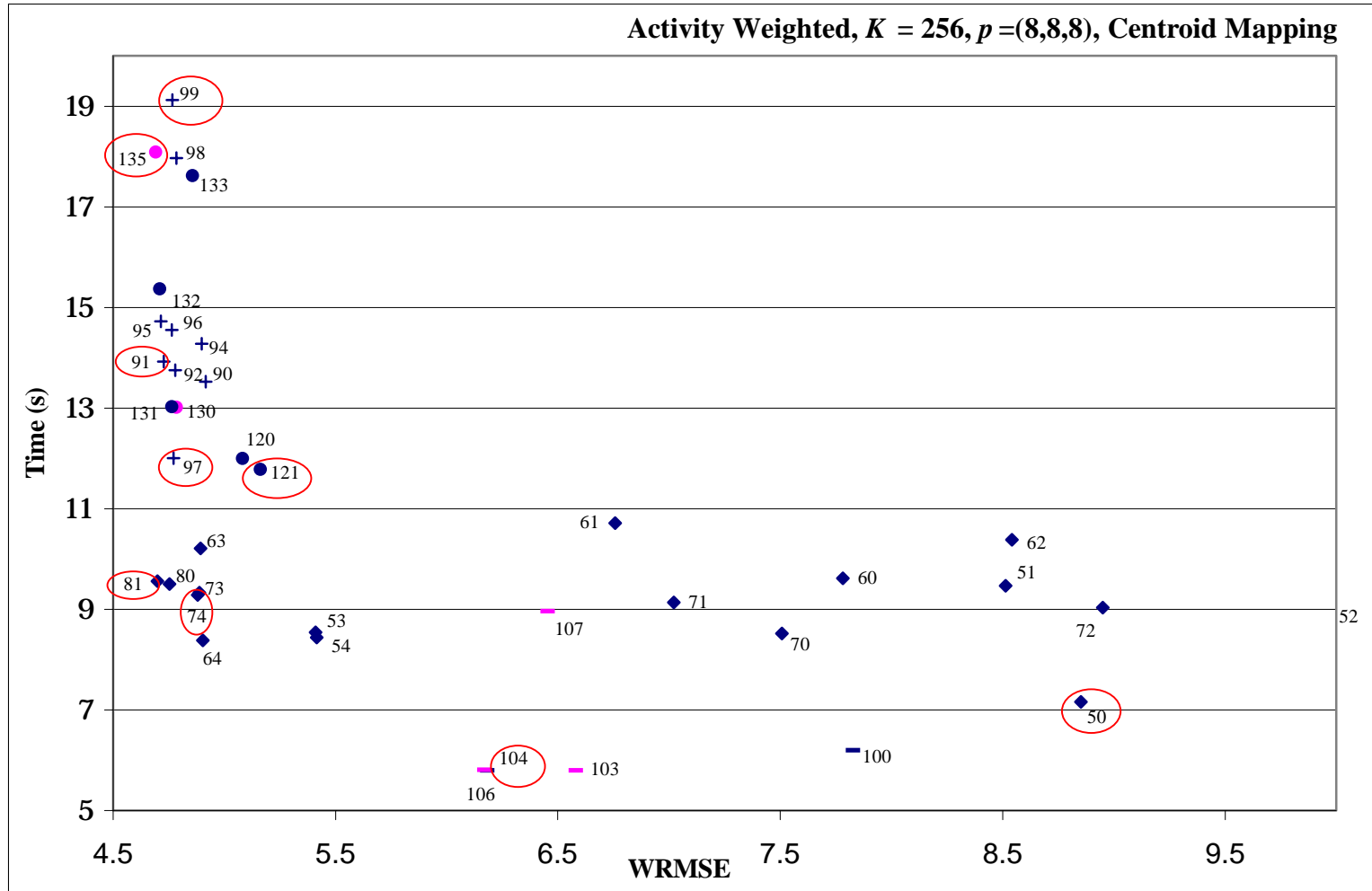


Figure 3.9: Average Running Time and WRMSE of Activity Weighted Divisive Algorithms. $K=256, p=(8,8,8)$.

#	Name	N	Time (s)	RMSE	WRMSE
1	<i>Shufelt</i>	20,302	1,671	4.54	2.32
2	<i>RGBCube</i>	34,111	5,570	8.12	8.87
3	<i>Shapes</i>	9,554	408	4.50	3.71
4	<i>Crambin</i>	19,073	1,770	5.69	4.50
5	<i>Solids</i>	31,429	5,260	3.66	2.72
6	<i>Marble</i>	21,532	2,502	4.20	3.65
7	<i>Cathedral</i>	28,662	4,590	5.21	3.96
8	<i>Sunset</i>	19,306	2,282	3.64	3.46
9	<i>Empire</i>	57,388	18,632	4.63	2.75
10	<i>Jhonni</i>	67,085	25,905	5.61	4.94
11	<i>Parrot</i>	40,565	9,085	4.34	3.28
12	<i>Matches</i>	64,112	23,087	4.85	4.52
13	<i>Mom</i>	96,486	52,888	7.68	5.97
14	<i>Tiger</i>	73,939	29,497	5.77	4.35
15	<i>Flowers</i>	71,687	30,284	6.26	5.61
16	<i>Peppers</i>	111,344	72,244	7.67	6.57
17	<i>Boys</i>	23,676	4,161	2.19	1.99
18	<i>Windsails</i>	86,008	41,089	7.20	5.25
19	<i>Cat</i>	95,916	55,485	3.95	3.38
20	<i>Woman</i>	94,279	50,166	7.50	5.81
21	<i>Lena</i>	148,279	132,201	5.96	5.26
22	<i>Boy</i>	180,274	211,870	5.27	4.47
23	<i>Eye</i>	267,246	461,443	6.56	6.05
24	<i>Chapel</i>	266,764	369,642	9.02	7.15
25	<i>Sailing</i>	177,844	209,299	4.69	3.97
	Max	267,246	461,443	9.02	8.87
	Min	9,554	408	2.19	1.99
	Avg	84,274	72,841	5.55	4.58

Table 3.4: **Running Time, RMSE, and WRMSE for Activity Weighted PNN.** $K=256$, $p=(8,8,8)$.

3.4 Subjective Observations

As previously stated in Section 2.5, WRMSE and RMSE are not the definitive measures of subjective image quality. Therefore, in this section the author discusses her subjective evaluation of a small subset of the quantized images analyzed in the previous section. In

addition, a small subset of the quantized images produced by the empirical analysis will be presented so that the readers may observe the image quality for themselves.

The author visually examined all images, $K=256$, $p=(8,8,8)$, and $\text{Select} \in \{\mathbf{P8C_D}, \mathbf{R2RM}\}$ and found that the oct-cut and non-variance-based median-cut algorithm produced quantized images with generally poor image quality (see Color Plate 3.4 and Color Plate 3.5). This subjective finding validates the empirical data presented in Figure 3.8.

The author visually examined all images, $K=256$, $p=(8,8,8)$, and $\text{Select} \in \{\sigma^2 2\sigma_x^2 \mu, \sigma^2 2\sigma_x^2 t, \sigma^2 2E_{\downarrow t}, \sigma^2 2\sigma_x^2 \perp, E_{\downarrow 2} E_{\downarrow \perp}, \sigma^2 2\mu_R \mu, \sigma^2 2e_1 \perp\}$ and generally found them to be quite good. For some of the “easy” images in the test set (*Shufelt*, *Marble*, *Cathedral*, *Empire*, *Tiger*, *Boys*, *Cat*, *Chapel* and *Sailing*), all of these algorithms produced quantized images with outstanding quality. On the rest of the images, which algorithm produced the highest quality image was image dependent, but the differences in quality were generally small. However, the author tends to favor the quantized images produced by $E_{\downarrow 2} E_{\downarrow \perp}$ and $\sigma^2 2e_1 \perp$, (99 and 135) which, of course, happen to be the slowest divisive algorithms studied.

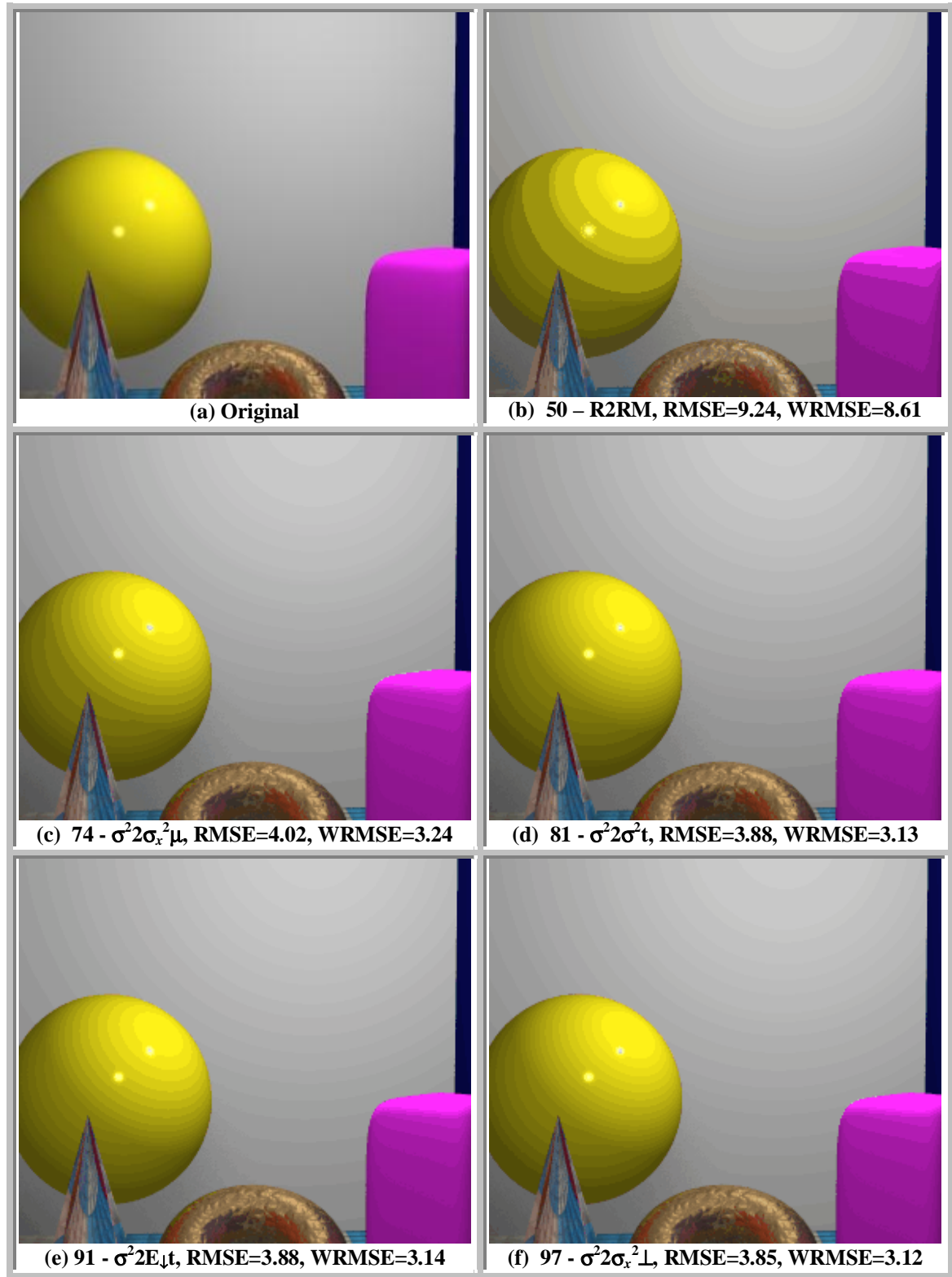
The author visually examined all images, $K=256$, $p=(8,8,8)$, and $\text{Select} \in \{E_{\downarrow 2} E_{\downarrow \perp}, \sigma^2 2\mu_R \mu, \mathbf{PNN}\}$ and generally found them to be of comparable excellent image quality. However, in a few cases, the author felt PNN did a noticeably better job. However, in no way did the author feel that the increase in image quality justified the enormous amount of time required by the PNN algorithm.

Color Plate 3.4 and Color Plate 3.5 illustrate the types of image quality differences produced by PNN and the algorithms circled in Figure 3.8. Of the images in the test set, *Solids* is one of the hardest to quantize. *Solids* is hard to quantize because it contains large smoothly shaded regions that are susceptible to contouring. Notice that even $E_{\downarrow 2} E_{\downarrow \perp}$ and $\sigma^2 2e_1 \perp$ had a difficult time quantizing *Solids*. While *Lena* also contains large smoothly varying regions, many of the colors in these regions occur a fairly compact region of RGB color space. On the other hand, *Solids* has smoothly varying regions from several diverse regions of the RGB color space (see Color Plate 2.2).

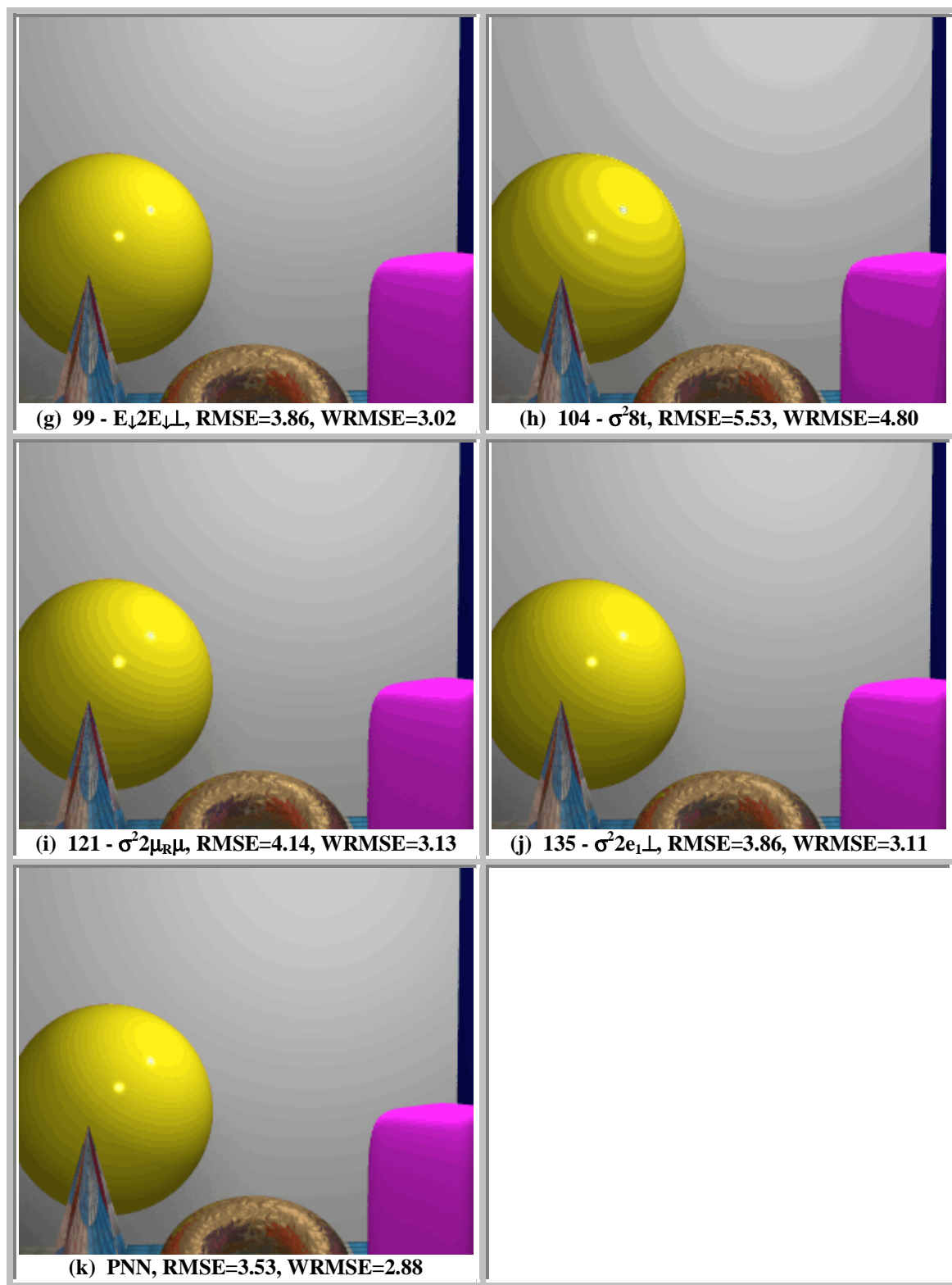
Next the effect activity weighting has on quantization time and image quality is examined. The author visually examined each activity weighted and non-activity weighted quantized image for $I \in \{Shufelt, RGBCube, Shapes, Crambin, Solids, Sunset, Jhonni, Parrot, Matches, Flowers, Windsails, Woman, Lena\}$, $K=256$, $p=(8,8,8)$, and $Select \in \{\sigma^2 2\sigma_x^2 \mu, \sigma^2 2\sigma_x^2 t, \sigma^2 2E_{\downarrow} t, \sigma^2 2\sigma_x^2 \perp, E_{\downarrow} 2E_{\downarrow} \perp, \sigma^2 2\mu_R \mu, \sigma^2 2e_1 \perp, PNN\}$ and nearly always found that the image quality was unchanged or slightly improved by activity weighting. Comparing the corresponding images in Color Plate 3.4, Color Plate 3.5, Color Plate 3.6 and Color Plate 3.7 illustrates this point. The arrows next to the RMSE and WRMSE in Color Plate 3.6 and Color Plate 3.7 indicate whether the measure increased or decreased with activity weighting. Notice that the RMSE always increased while WRMSE always decreased. Thus, WRMSE seems to be a better measure of visual image quality than RMSE.

3.5 Summary

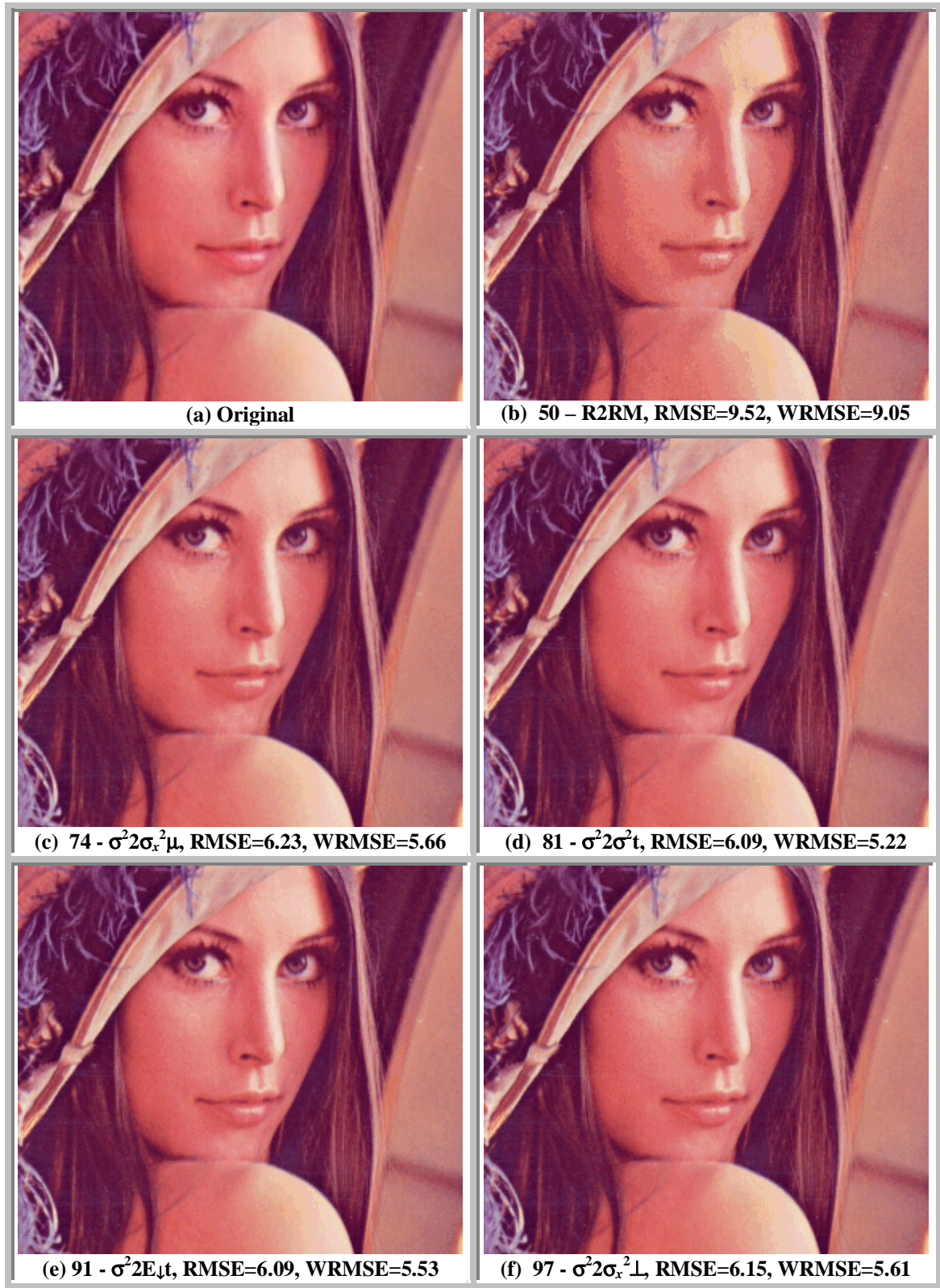
This chapter conducted a comprehensive survey and comparative analysis of PNN and hierarchically divisive color quantization techniques. PNN was shown to achieve quantized images with lower RMSE than the divisive algorithms; however, PNN requires an impractical amount of time to do so. Several of the divisive methods had not been previously studied. The divisive algorithms were discussed within the framework of a general taxonomy for classifying divisive algorithms proposed by this dissertation. Section 5.1 proposes the heterogeneous-cut algorithm that combines with the speed advantages of oct-cut methods and the accuracy of variance-minimization and principal component oblique-cut techniques. The heterogeneous-cut algorithm obtains a position on Figure 3.8 and Figure 3.9 that is the closest to origin. That is, it obtains the low error very quickly.



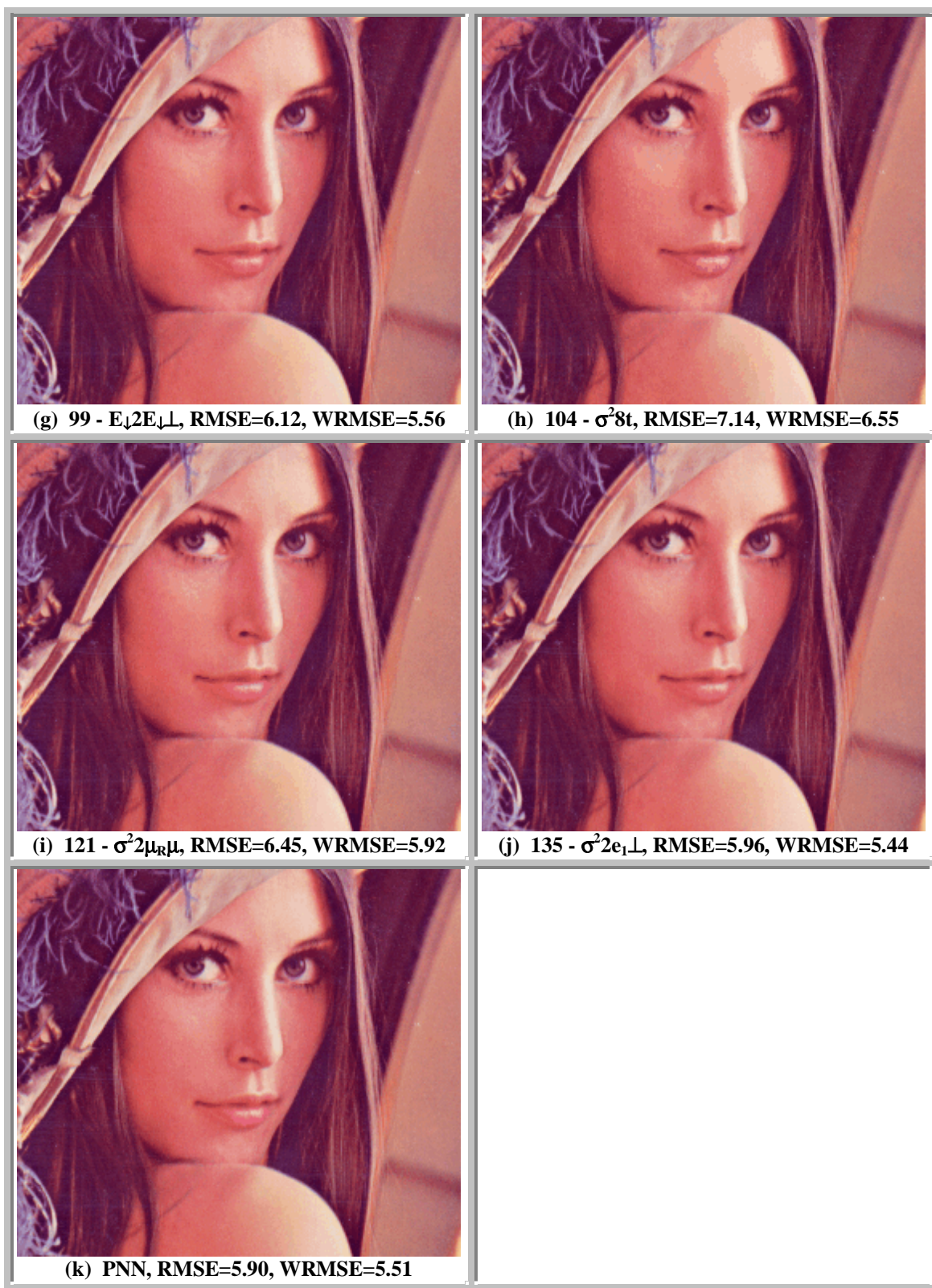
Color Plate 3.4: **Hierarchical Quantization of Solids.** $K=256$, $p=(8,8,8)$.



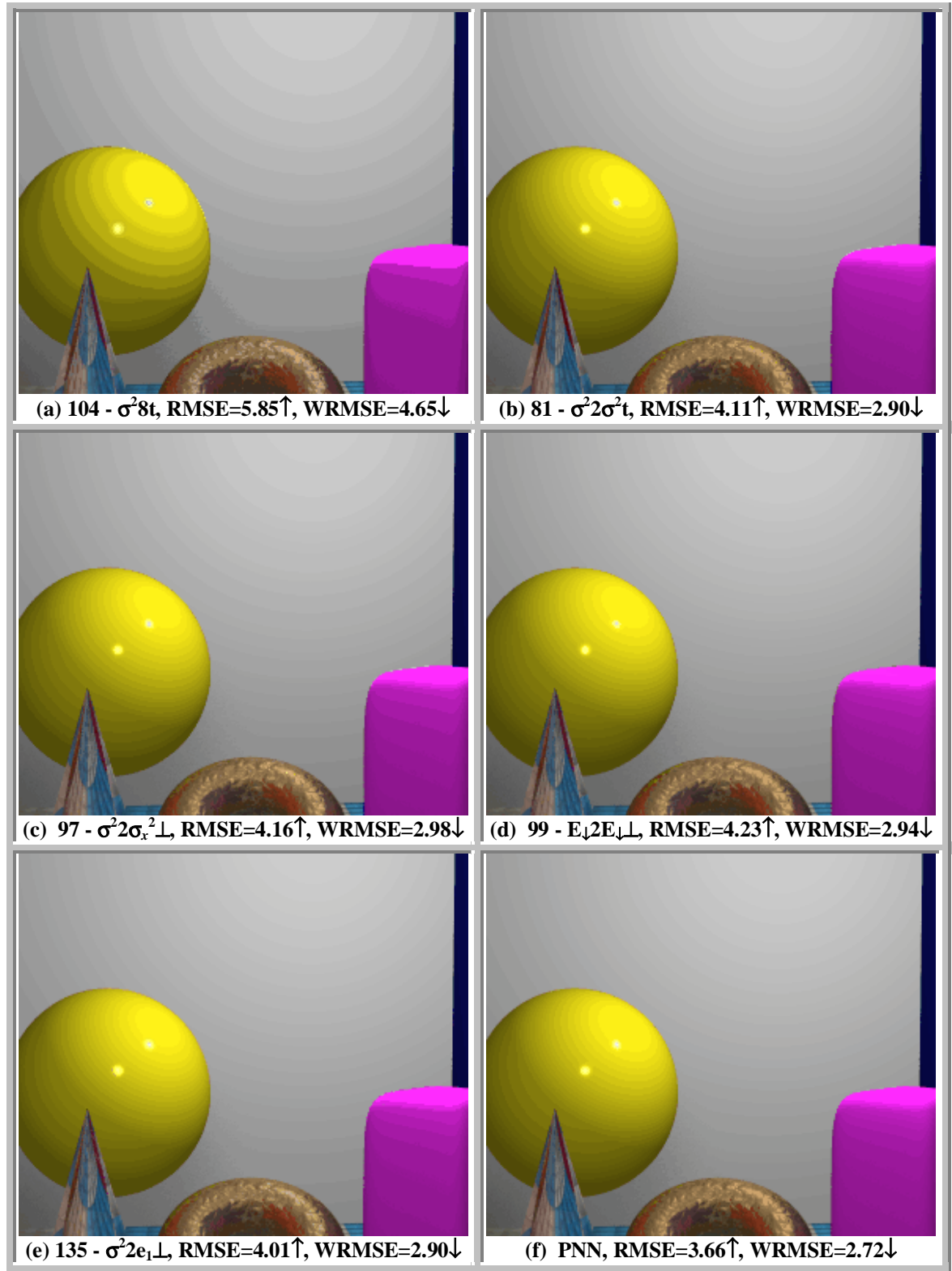
Color Plate 3.4: **Continued.**



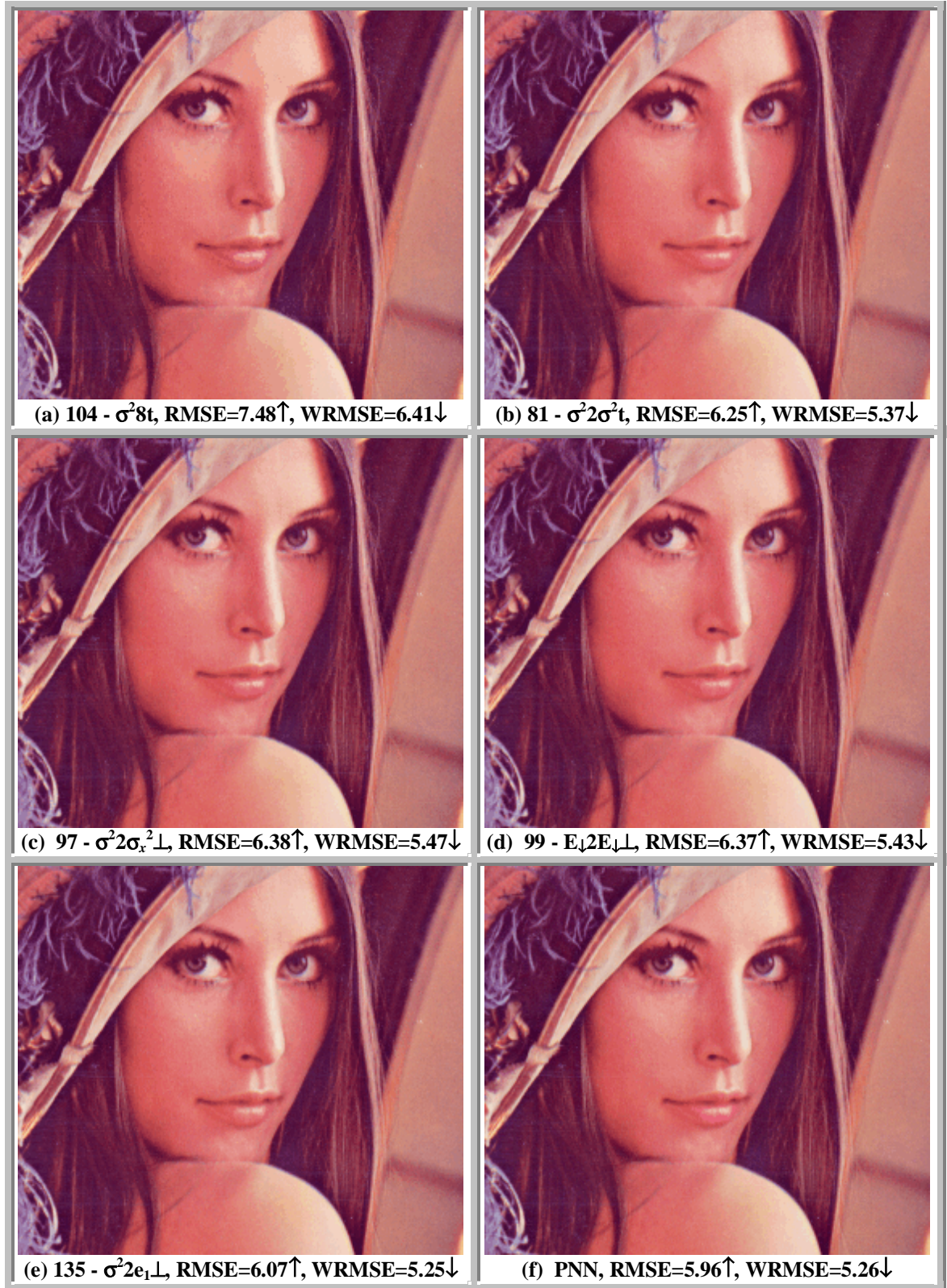
Color Plate 3.5: **Hierarchical Quantization of *Lena*.** $K=256$, $p=(8,8,8)$.



Color Plate 3.5: **Continued.**



Color Plate 3.6: Activity Weighted Hierarchical Quantization of *Solids*. $K=256$, $p=(8,8,8)$.



Color Plate 3.7: Activity Weighted Hierarchical Quantization of *Lena*. $K=256$, $p=(8,8,8)$.

Chapter 4

Non-Hierarchical Methods

This chapter describes several non-hierarchical methods for color image quantization. The bulk of the non-hierarchical methods can be lumped into the broad category of iterative refinement. These algorithms begin with some initial set of representative colors and then alter cluster memberships so as to obtain a better partition. The improvement is achieved by lifting the permanence of decisions made by hierarchical algorithms of whether to merge in the case of agglomerative or to divide in the case of divisive. In many cases, the quality of the quantized image is largely determined by the initial set of representative colors used by the iterative refinement method. This chapter describes the Linde-Buzo-Gray (LBG) algorithm as its one example of an iterative refinement method [LiBG80].

Non-hierarchical methods that methods that will not be described by this dissertation include:

- (i) *K*-means [KoCL93 and VeB95],
- (ii) simulated annealing [FiO89],
- (iii) neural networks [Dek94, GaMC94, ReSMG97a, ReSMG97b, and PeL98], and
- (iv) evolutionary or genetic programming [Sch96a, FrS97, and TaAE98].

The main problem with these techniques is that they have several parameters to which their performance is very sensitive. A couple of hybrid algorithms have also been proposed:

- (i) genetic and *K*-means [Sch97],
- (ii) genetic and LBG [Sch96b]

In the future it would be interesting to compare the methods proposed in Chapter 5 to some of these non-hierarchical methods.

4.1 The Popularity Algorithm

This section takes a quick look at the popularity algorithm [Bra86, Hec80 and Cla95]. The popularity algorithm is of interest primarily because it was the first color image quantization algorithm proposed. This algorithm simply selects the K *most popular* colors in the truecolor image; that is, the K colors with the highest frequency are chosen. The popularity algorithm works best if the histogram has been prequantized using bit-cutting, thus clustering neighboring colors in the histogram.

Color Plate 4.1 illustrates typical image quality produced by the popularity algorithm. Notice that images quantized $R_5G_6B_4$ look significantly better than those quantized in $R_8G_8B_8$. The popularity algorithm can also be modified to incorporate activity weighting; that is, the K colors with the highest activity weighting are chosen. Color Plate 4.1(c) and (d) show how activity weighting can effect the quality of quantized images produced by the popularity algorithm. In *Solids*, the yellow ball has become nearly flawless, but contouring in the white wall and purple cube has become much more severe. Activity weighting has clearly improved the image quality in *Lena*, despite the fact that the WRMSE for the activity weighted image is higher than the non-activity weighted image.

The primary drawback of the popularity algorithm is that sparse, but potentially important areas of the histogram are often under-represented. In order to ensure a better distribution among the selected representatives, several modifications to the basic popularity algorithm have been proposed. For example, after a color is selected, Braudaway [Bra86] reduces the counts of neighboring colors in the histogram using a spherically symmetric exponential function². Clark [Cla95] divides the histogram into four regions (red-dominant, green-dominant, blue-dominant, and gray-dominant) and then selects one-fourth of representative colors from each of the four regions. These enhancements are not further explored by this dissertation.

² Braudaway's technique is patented, United States Patent No. 4,907,075.



Color Plate 4.1: **Popularity Quantization of Solids and Lena.** $K=256$.

4.2 The MinMax Algorithm

In 1985, Gonzalez [Gon85] proposed an approximation algorithm that seeks to minimize the maxDiam metric defined in Section 2.4.4. In 1986, Houle and Dubois [HoD86] proposed the MinMax algorithm for color image quantization which turns out to be Gonzalez's general purpose clustering algorithm applied to color image quantization. The goal of the MinMax algorithm is to obtain a K -split $S^* = \{s_0^*, s_1^*, \dots, s_{K-1}^*\}$ for C such that

$$\text{maxDiam}(S^*) = \min\{\text{maxDiam}(S) \mid S \text{ is a } K\text{-split for } C\}. \quad (4.1)$$

Pseudocode for the MinMax algorithm is provided in Figure 4.1.

Algorithm:	MinMax algorithm to select representatives
Input: C	The set of N unique colors in I
K	The numbers of colors to be selected.
	<i>Assumption:</i> $N > K$.
Output: R_K	Selected representatives for I
S_K	K -split for C

```

// Step 1: Initialization
1   $s_0 \leftarrow C = \{c_1, c_2, \dots, c_N\} \subseteq \text{RGB}$ 
2   $r_0 \leftarrow c_1$ 
// Step 2: Iterative Creation of New Clusters
3  for  $k \leftarrow 1$  upto  $K-1$ 
4     $\delta \leftarrow \max \{d_2(c_i, r_j) \mid c_i \in s_j \text{ and } 0 \leq j \leq k-1\}$ 
5     $c \leftarrow$  one of the colors whose distance to its representative is  $\delta$ 
6    move  $c$  to  $s_k$ 
7     $r_k \leftarrow c$ 
// Step 3: Reassignment of Colors to New Cluster
8    for each  $j \leftarrow 0$  upto  $k-1$ 
9      for each  $c \in s_j$ 
10       if  $d_2(c, r_k) \leq d_2(c, r_j)$ 
11         move  $c$  to  $s_k$ 
12  return  $R_K$  and  $S_K$ 

```

Figure 4.1: Pseudocode for the MinMax Algorithm.

4.2.1 Theoretical Interest

One of the interesting aspects of the MinMax algorithm is that Gonzalez [Gon85] proved it constructs a K -split whose $maxDiam$ is less than or equal to two times the $maxDiam$ of the optimal K -split. His proof is formalized in Lemma 4.1 and Theorem 4.1.

Lemma 4.1: $MinMax \Rightarrow \max_{\substack{c_j \in s_i \\ s_i \in S_K}} d_2(c_j, r_i) \leq \min_{\substack{l \neq m \\ r_l, r_m \in R_K}} d_2(r_l, r_m).$

Let C be the set of N unique colors in truecolor image I . Let $K < N$. Let $S_K = \{s_0, s_1, \dots, s_{K-1}\}$ be the K -split for C obtained by the MinMax algorithm. Likewise, let $R_K = \{r_0, r_1, \dots, r_{K-1}\}$ be the representative colors obtained by the MinMax algorithm.

Then $\max_{\substack{c_j \in s_i \\ s_i \in S_K}} d_2(c_j, r_i) \leq \min_{\substack{l \neq m \\ r_l, r_m \in R_K}} d_2(r_l, r_m).$

Proof: By loop induction on k .

Basis. Let $k=1$, then $S_2 = \{s_0, s_1\}$ and $R_2 = \{r_0, r_1\}$. By lines 4-7 of Figure 4.1, $\delta = d_2(r_0, r_1)$. By line 4, $\forall c \in C, d_2(c, r_0) \leq \delta$. If $c \in s_1$ then by lines 8-11 of Figure 4.1, $d_2(c, r_1) \leq \delta$. Thus $\max_{\substack{c_j \in s_i \\ s_i \in S_2}} d_2(c_j, r_i) \leq \min_{\substack{l \neq m \\ r_l, r_m \in R_2}} d_2(r_l, r_m).$ ■

Induction. Let $k = K-1$, then $S_K = \{s_0, s_1, \dots, s_{K-1}\}$ and $R_K = \{r_0, r_1, \dots, r_{K-1}\}$. Observe that a color does not change cluster membership unless its distance to the new cluster representative is less than the current distance to its representative and that cluster representatives never change. Thus, $\max_{\substack{c_j \in s_i \\ s_i \in S_K}} d_2(c_j, r_i) \leq \max_{\substack{c_j \in s_i \\ s_i \in S_{K-1}}} d_2(c_j, r_i).$

However, by the induction hypothesis and lines 4-7 of Figure 4.1,

$\max_{\substack{c_j \in s_i \\ s_i \in S_{K-1}}} d_2(c_j, r_i) = \min_{\substack{l \neq m \\ r_l, r_m \in R_K}} d_2(r_l, r_m).$ Thus, $\max_{\substack{c_j \in s_i \\ s_i \in S_K}} d_2(c_j, r_i) \leq \min_{\substack{l \neq m \\ r_l, r_m \in R_K}} d_2(r_l, r_m).$ ■

Theorem 4.1: $\text{MinMax} \Rightarrow \maxDiam(S_K) \leq 2 \cdot \maxDiam(S_K^*)$.

Let C be the set of N unique colors in truecolor image I . Let $K < N$. Let $S_K = \{s_0, s_1, \dots, s_{K-1}\}$ be the K -split for C obtained by the MinMax algorithm. Likewise, let $R_K = \{r_0, r_1, \dots, r_{K-1}\}$ be the representative colors obtained by the MinMax algorithm. Let $S_K^* = \{s_0^*, s_1^*, \dots, s_{K-1}^*\}$ be an optimal K -split for C . That is, $\maxDiam(S_K^*) = \min\{\maxDiam(S) \mid S \text{ is a } K\text{-split for } C\}$. Then $\maxDiam(S_K) \leq 2 \cdot \maxDiam(S_K^*)$.

Proof: By construction.

Suppose the algorithm in Figure 4.1 picks one more representative r_K . Let $\delta = \max_{\substack{c_j \in s_j \\ s_i \in S_K}} d_2(c_j, r_i)$. Then by Lemma 4.1, $\min_{\substack{l \neq m \\ r_l, r_m \in R_K \cup r_k}} d_2(r_l, r_m) \geq \delta$. That is, the

minimum pairwise distance between representatives is at least δ . In any K -split of C at least two of the representatives in $R_K \cup r_k$ must be in the same cluster. Hence, δ is a lower bound on the maximum diameter of S_K^* . That is, $\maxDiam(S_K^*) \geq \delta$. However, it has already been established that $\delta =$

$\max_{\substack{c_j \in s_j \\ s_i \in S_K}} d_2(c_j, r_i)$. Thus by the triangle inequality, the $\maxDiam(S_K) = 2 \cdot \delta$.

Hence, $\maxDiam(S_K) \leq 2 \cdot \maxDiam(S_K^*)$. ■

The application of the MinMax algorithm to color image quantization has been studied and enhanced by several researchers [HoD86, Gol91, Xia97 and TrCC97]. For example, to speed up the MinMax algorithm given in Figure 4.1, both Houle and Dubois [HoD86] and Goldberg [Gol91] proposed using a subset C' of the entire set C of N unique colors in I :

$$C' = \{c' \mid H(c') \geq v_c\}, \quad (4.2)$$

where v_c is a predetermined threshold [HoD86 and Gol91]. Clearly when $v_c = 0$, $C' = C$. In Gonzalez's algorithm [Gon85], the selection of r_0 is arbitrary. Houle and Dubois [HoD86] suggest using either the most popular color, or black as the initial

representative. The rationale for using black is based on their claim that black is one of the most probable colors found in natural images.

In 1997, without any reference to Houle and Dubois [HoD86] or Goldberg [Gol91], Xiang [Xia97] proposed the use of Gonzalez's algorithm [Gon85] for color image quantization. Xiang suggested two modifications to the MinMax algorithm:

- (i) the centroid of each of the K clusters produced in the for loop in lines 3-11 of Figure 4.1 become the representative colors for each of the K clusters, and
- (ii) the distance function is scaled to account for the relative importance of each of the R, G, and B channels:

$$d_s(x, y) = \sqrt{0.5 \cdot (x.R - y.R)^2 + (x.G - y.G)^2 + 0.25 \cdot (x.B - y.B)^2}. \quad (4.3)$$

These two changes do not invalidate Theorem 4.1 because the diameters of the final clusters remain unchanged, and because the scaled distance function is metric. The next two sections show that these two changes help to improve the quality of the quantized images; however, the quality is far inferior to some of the better hierarchical methods discussed in the previous chapter.

4.2.2 Empirical Analysis

To determine the merits of the *maxDiam* metric, this dissertation examines three versions of the MinMax algorithm. The purpose of examining the MinMax algorithm is to determine the potential accuracy to achievable by the MinMax algorithm. Therefore, all versions let $v_c=0$, and all experimental results are for $K=256$ and $p=(8,8,8)$. Since black is not necessarily a color found in all images, nor is it necessarily one of the most probable colors, all versions choose the most popular color as the initial representative.

The first version of MinMax implements the original algorithm. Version two is identical to the first version except the first modification suggested by Xiang [Xia97] has been implemented. That is, the centroids of the resulting clusters are used as the K representative colors. The third version of MinMax is identical to the second version except the scaled distance function (d_s) given by Equation 4.3 is used.

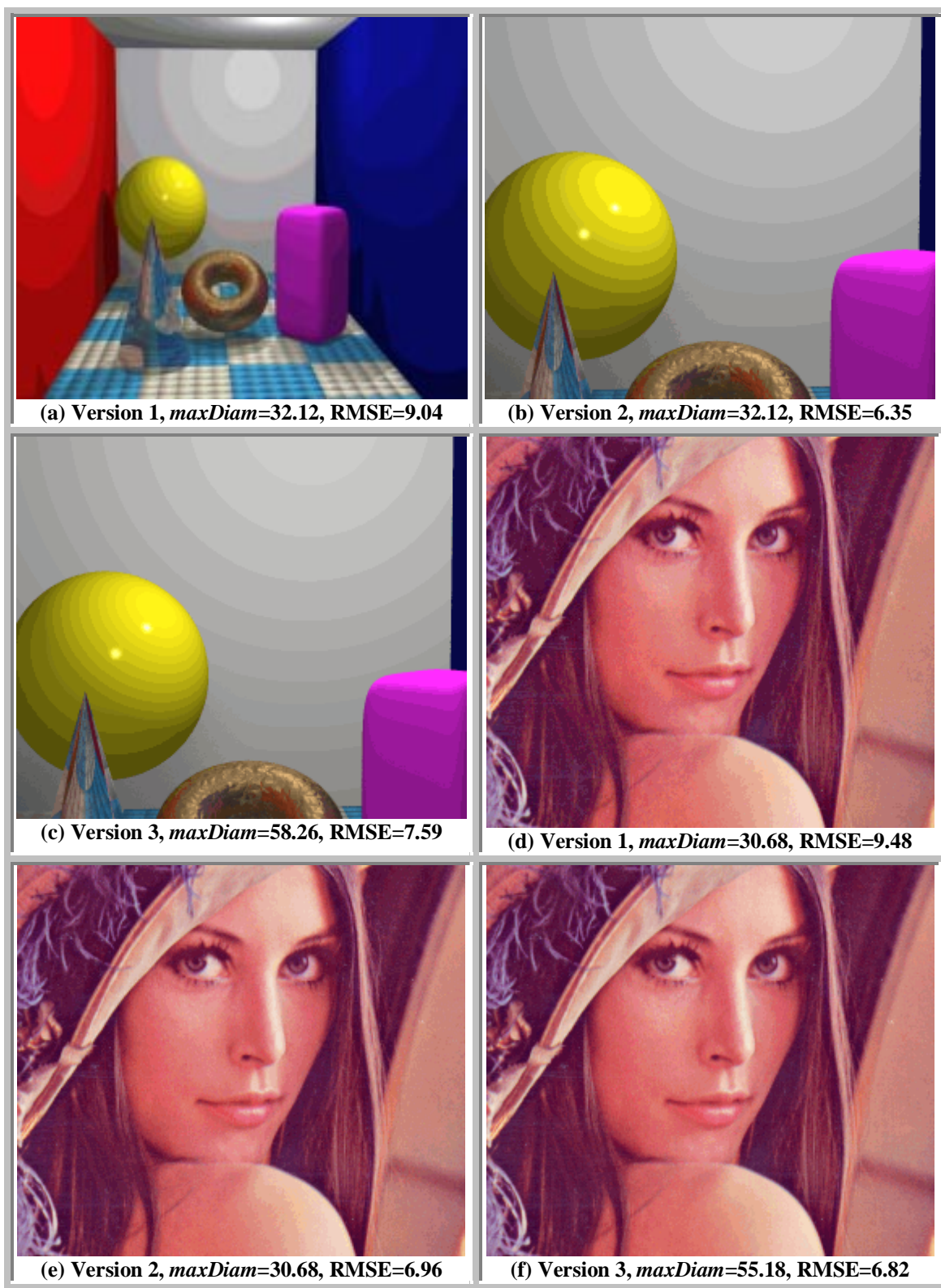
<i>Version</i>	<i>Time (s)</i>	<i>maxDiam</i>	<i>RMSE</i>
1	57.62	33.69	9.32
2	56.81	33.69	7.03
3	57.48	63.65	7.17

Table 4.1: **Average Running Time, *maxDiam*, RMSE, and WRMSE for MinMax.**
 $K=256, p=(8,8,8)$.

Table 4.1 gives the average running time, *maxDiam*, RMSE, and WRMSE for each of the three versions of MinMax for $K=256$ and $p=(8,8,8)$. The average running time of MinMax is about three times greater than the slowest hierarchically divisive algorithm analyzed in Section 3.3. One would hope that this increase in running time would lead to quantized images with lower RMSE. Unfortunately, MinMax doesn't deliver. Instead, the average RMSE for MinMax is comparable to the oct-cut and non variance-based median-, center-, and mean-cut algorithms analyzed in Section 3.3. Thus, MinMax spends about ten times longer than the oct-cut and non variance-based median-, center-, and mean-cut algorithms, while producing quantized images with comparable quality in terms of RMSE. Note that the *maxDiam* for the third version is about twice that of the other two versions. This is because the third version uses the scaled distance function (d_s , Equation 4.3) while *maxDiam* uses the Euclidean distance function (d_2 , Equation 2.11).

4.2.3 Subjective Observations

Color Plate 4.2(a) and (d) show the results of quantizing *Lena* and *Solids* using version one of the MinMax algorithm. As you can see, the results are very disappointing due to the considerable contouring in the numerous smooth regions of the images. In addition to contouring, there is considerable color shifting in *Solids*. Color Plate 4.2 (b) and (e) show that using the centroids of the resulting clusters for the K representative colors yields some improvement; however, the contouring remains quite unsatisfactory. Color Plate 4.2 (c) and (f) show that the scaled distance function helps to improve the image quality of *Lena*, but that in the case of *Solids*, the image quality remains quite poor.



Color Plate 4.2: **MinMax Quantization of Solids and Lena.** $K=256$, $p=(8,8,8)$.

As previously noted, the *maxDiam* for the third version is about two times greater than the *maxDiam* for the first two versions. However, the quantized images produced by the third version do not look significantly worse than the images produced by the first two versions. Thus, for the purposes of color image quantization, *maxDiam* (using the Euclidean distance function) is not a reasonable metric. For Color Plate 4.2, RMSE reasonably measures the subjective image quality.

4.3 Linde-Buzo-Gray (LBG)

Algorithm:	LBG algorithm to select representatives
Input:	C The set of N unique colors in I
	K The numbers of colors to be selected.
	<i>Assumption:</i> $N > K$.
	H Histogram of I
	R_K Initial representatives for I
	<i>threshold</i> Minimum number of pixels that can change clusters
Output:	R_K Selected representatives for I
	S_K K -split for C
<i>// Step 1: Initialization</i>	
1	for $j \leftarrow 1$ upto N
2	add $c_j \in C$ to cluster s_k s.t. $d_2(c, r_k) \leq d_2(c, r_l), 0 \leq l < K$
3	for $j \leftarrow 0$ upto $K-1$
4	$r_j \leftarrow$ centroid of s_j
<i>// Step 2: Iterative Reassignment of Colors to Clusters</i>	
5	do
6	for $j \leftarrow 0$ upto $K-1$
7	for each $c \in s_j$
8	move c to cluster s_k s.t. $d_2(c, r_k) \leq d_2(c, r_l), 0 \leq l < K$
9	for $j \leftarrow 0$ upto $K-1$
10	$r_j \leftarrow$ centroid of s_j
11	until the percentage of pixels that changed clusters \leq <i>threshold</i>
12	return R_K and S_K

Figure 4.2: Pseudocode for the LBG Algorithm.

The *Linde-Buzo-Gray* (LBG) algorithm is a fixed-point iterative refinement algorithm developed by [LiBG80] for vector quantization codebook design. Pseudocode for the

LBG algorithm is provided in Figure 4.2. When the *threshold* parameter used in line 12 of Figure 4.2 is zero, the LBG algorithm is guaranteed to converge to a local minimum; however, there is no guarantee that it will converge to an optimal solution. Not surprisingly, the fidelity the LBG algorithm depends heavily upon the initial K representatives. As such, the LBG algorithm is often used as a post-process to refine the K representatives selected by some other method. The LBG algorithm studied in the following two sections begins with initial representatives that were chosen by the $\sigma^2 \mathbf{2e}_1 \perp_{\mathbf{L}_e}$ hierarchically divisive algorithm. The threshold used was zero. That is, LBG was run until it converged at a local minimum.

4.3.1 Empirical Analysis

#	<i>Time</i> (s)	<i>Time</i> _{LBG} (s)	<i>Iterations</i>	<i>RMSE</i>	<i>WRMSE</i>
Max	709.18	654.20	41	8.61	8.57
Min	17.69	11.26	8	1.65	1.51
Avg	166.29	147.40	19.0	5.10	4.41
Max	618.02	564.36	38	8.69	8.58
Min	22.44	14.93	9	1.81	1.49
Avg	179.00	159.39	19.4	5.27	4.25

Table 4.2: **Empirical Data for LBG.** $K=256$, $p=(8,8,8)$, *threshold*=0. The initial representatives were chosen using the $\sigma^2 \mathbf{2e}_1 \perp_{\mathbf{L}_e}$ hierarchically divisive algorithm. The shaded data indicates activity weighting was used.

Table 4.2 provides empirical data for the LBG algorithm where the initial representatives were selected using the $\sigma^2 \mathbf{2e}_1 \perp_{\mathbf{L}_e}$ hierarchically divisive algorithm. The time listed in the first column represents the total running time including selecting the initial representatives. The time listed in the second column represents time to improve the initial representatives using LBG. The fourth column gives the total number of iterations required by LBG to reach a local minimum (threshold = 0). Notice that the average RMSE and WRMSE for LBG is slightly lower than for PNN (Table 3.3 and Table 3.4), but that LBG requires less than 0.3% of the time on average.

4.3.2 Subjective Observations

Color Plate 3.5 illustrate the types of image quality differences produced by LBG when initialized with representatives chosen by an excellent algorithm ($\sigma^2 \mathbf{2e}_1 \perp_e$). As with the hierarchical algorithms observed in Section 3.4, activity weighting slightly enhances image quality. Notice that in Color Plate 3.5 activity weighting increases RMSE but decreases WRMSE. Thus, WRMSE seems to be a better measure of visual image quality than RMSE. The author visually examined all images, activity weighted, $K=256$, $p=(8,8,8)$, and $\text{Select} \in \{\sigma^2 \mathbf{2e}_1 \perp_e, \text{PNN}, \text{LBG}\}$ and generally found them to be of comparable excellent image quality. LBG enhanced the image quality of $\sigma^2 \mathbf{2e}_1 \perp_e$ quantized images in a manner similar to the improvements added by activity weighting, small and subtle. In a few cases the PNN algorithm did a noticeably better job than $\sigma^2 \mathbf{2e}_1 \perp_e$, but LBG helped to close the gap.

4.4 Summary

This chapter described and analyzed three non-hierarchical algorithms. The analysis of the MinMax algorithm showed that the *maxDiam* metric is inappropriate for color image quantization. The LBG iterative refinement algorithm was shown to be a much more cost-effective way to achieve image quality close to PNN. Section 5.1.1 proposes an algorithm that is a hybrid between the MinMax algorithm and LBG. The new algorithm incorporates frequency (or activity weighting) information in order to obtain quantized images with significantly less visual distortion than the MinMax algorithm. The running time of the new method is in between MinMax and LBG, but closer to MinMax.



Color Plate 4.3: **LBG Quantized Images.** Truecolor images *Lena* and *Solids* quantized to 256 colors using LBG iterative refinement, $p=(8,8,8)$. The initial representatives were chosen using the $\sigma^2 \mathbf{e}_1 \perp_{\mathbf{e}}$ hierarchically divisive algorithm.

Chapter 5

Proposed Methods

This chapter describes and analyzes two new color quantization techniques. Section 5.1 proposes the heterogeneous-cut algorithm that combines five of hierarchically divisive color image quantization algorithms to obtain high quality quantized images quickly. Section 5.2 proposes a hybrid between MinMax and LBG called the weighted MinMax method.

5.1 Heterogeneous-Cut Algorithm

The heterogeneous-cut algorithm proposed by this dissertation results from the systematic and comprehensive study of divisive algorithms conducted in Chapter 3. The heterogeneous-cut algorithm combines five of these divisive algorithms in a unique way that capitalizes on the strengths of each technique, but does not suffer from their weaknesses. A flow-chart for the proposed heterogeneous-cut algorithm is provided in Figure 5.1. The heterogeneous-cut algorithm always selects the partition with the largest variance to further subdivide. Let n be the number of unique colors in the partition to be divided. The decision as to the number of subpartitions, as well as the position and orientation of the partition planes depends how n compares to N and the percentages of variance in each of the three directions. Four threshold values are used (t_1, t_2, t_3, t_4). The first three thresholds are selected such that $t_1 < t_2 < t_3 < 0.05$. The empirical results reported in this dissertation use the following threshold values: $t_1=0.004$, $t_2=0.010$, $t_3=0.026$, $t_4=20$.

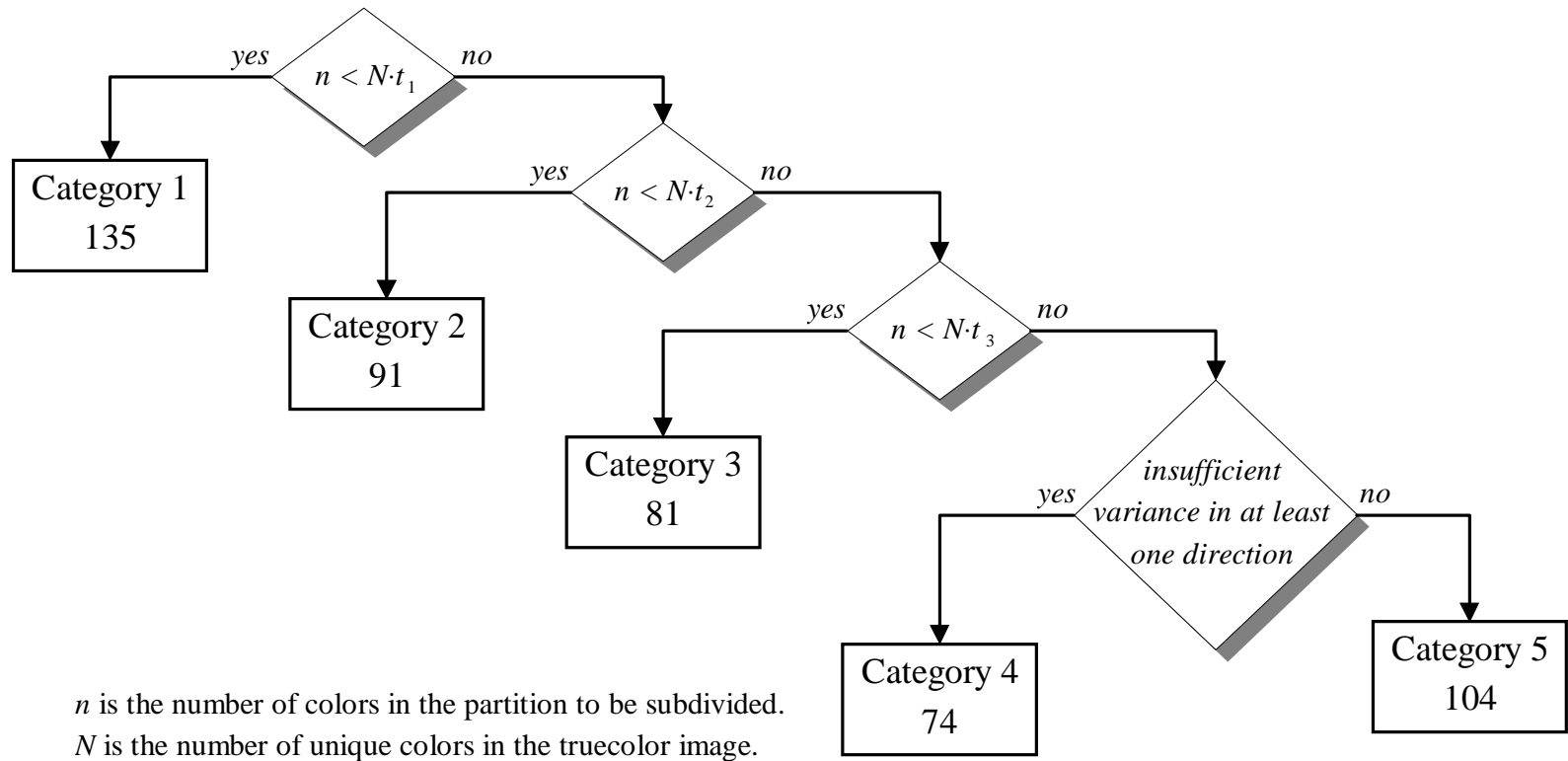


Figure 5.1: **Flow-chart for the Proposed Heterogeneous-Cut Algorithm.** The heterogeneous-cut algorithm combines five divisive algorithms in a unique way.

The time required to split a partition depends on n . The oct-cut algorithms obtain their speed by reducing the number of splits necessary because each split results in seven new subpartitions. The oct-cut algorithms produce low quality quantized images because partitions are often split in three directions when the partition doesn't have much variance in one or two of the directions. Therefore, the heterogeneous-cut algorithm uses the oct-cut algorithm ($\sigma^2 8t$, 104) only when n is large ($n \geq N \cdot t_3$) and when the percentage of variance in all three directions is larger than t_4 . Otherwise, when n is large ($n \geq N \cdot t_3$), but there is not sufficient variance in one or more of the directions, the variance-based mean split algorithm ($\sigma^2 2 \sigma^2 \mu$, 74) is used. This algorithm was selected because it is relatively fast and produces images with relatively low RMSE.

On the other hand, the oblique-cut algorithm ($\sigma^2 2 e_1 \perp e$, 135) produces high quality quantized images, but is slow. Therefore, this oblique-cut algorithm is used only n is small ($n < N \cdot t_1$). The heterogeneous-cut algorithm also incorporates two other algorithms ($\sigma^2 2 \sigma^2 t$ and $\sigma^2 2 E_{\downarrow} t$) that produce images with relatively low RMSE. Since $\sigma^2 2 E_{\downarrow} t$ (91) requires more time than $\sigma^2 2 \sigma^2 t$ (81), $\sigma^2 2 E_{\downarrow} t$ is used when n is medium small ($N \cdot t_1 \leq n < N \cdot t_2$) and $\sigma^2 2 \sigma^2 t$ is used when n medium large ($N \cdot t_2 \leq n < N \cdot t_3$).

5.1.1 Empirical Analysis

The success of the heterogeneous-cut algorithm is highlighted in Figure 5.2. This figure compares the average running time and RMSE of the proposed heterogeneous-cut algorithm (150) to the divisive algorithms listed in Table 3.2. The algorithms used by the heterogeneous-cut algorithm are circled in red. The heterogeneous-cut algorithm is surrounded by a star. The heterogeneous-cut algorithm is not much slower than the oct-cut algorithm (104), but produces quantized images with significantly less RMSE. For the given test set, one could justifiably call a quantization technique “high quality” if the average RMSE is less than 6.0. Given this definition of high quality, the heterogeneous-cut algorithm is the clear winner with respect to time.

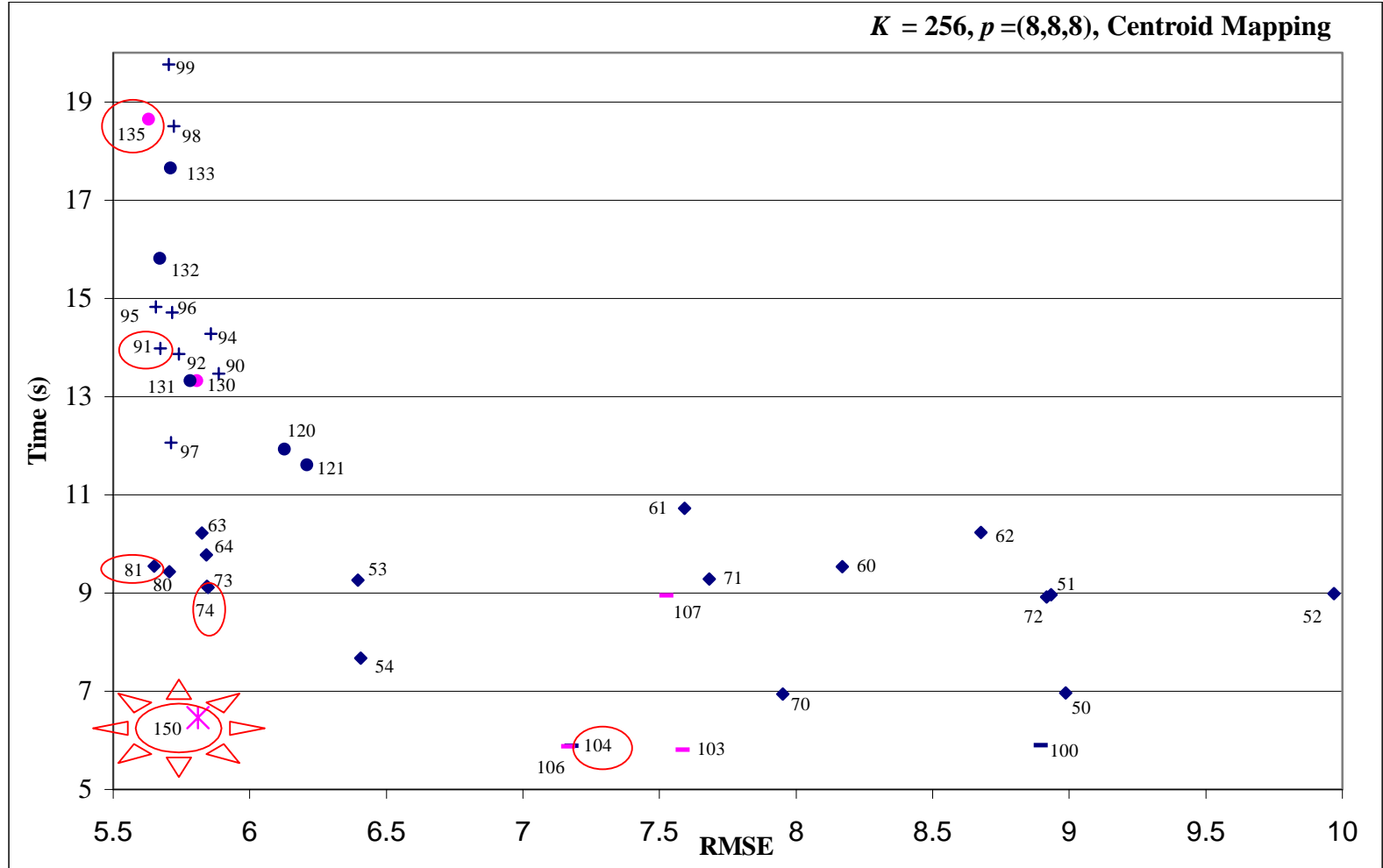


Figure 5.2: Average Running Time and RMSE for Proposed Heterogeneous-Cut Algorithm. $K=256, p=(8,8,8)$.

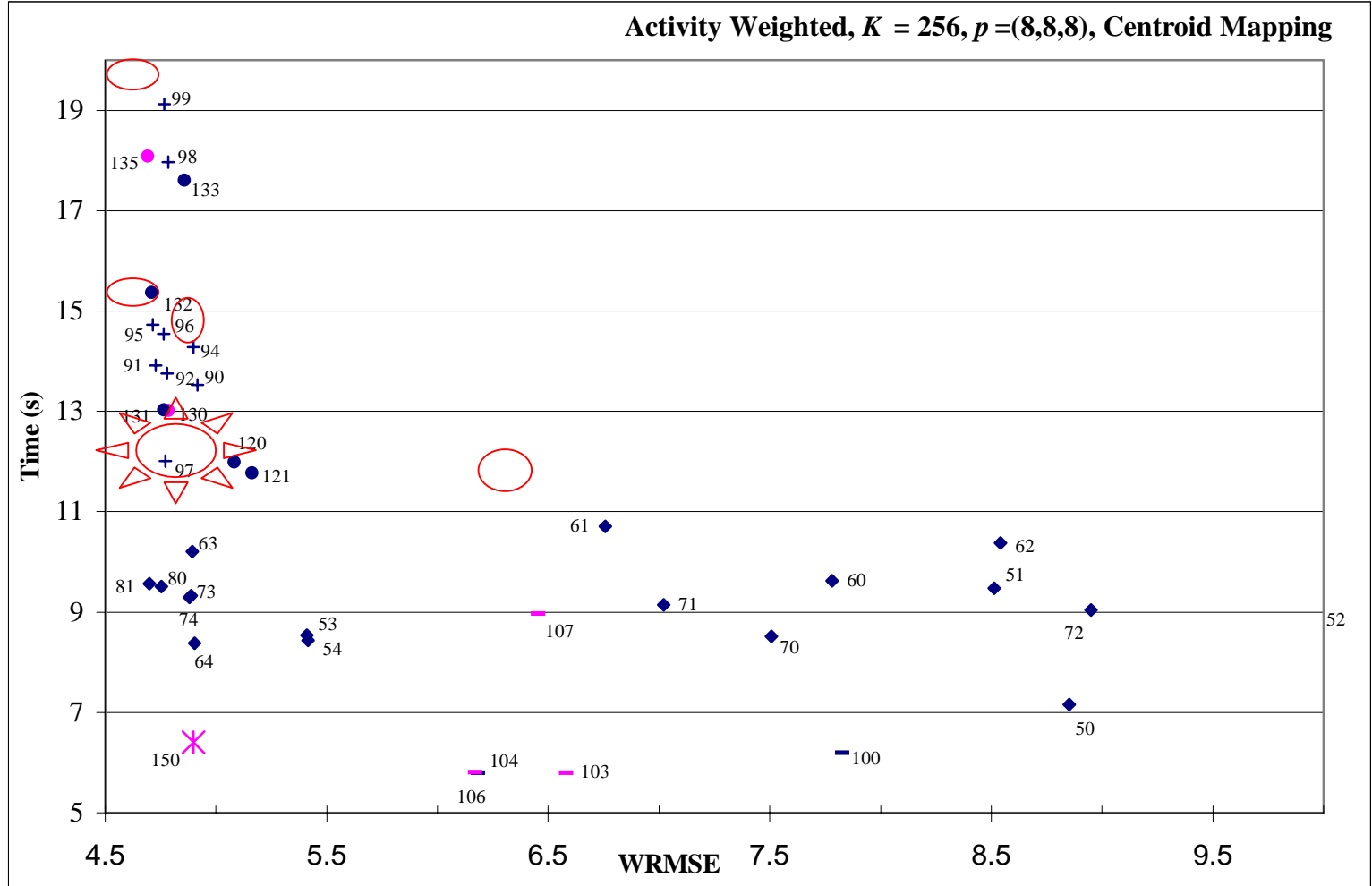


Figure 5.3: Average Running Time and WRMSE for Proposed Activity Weighted Heterogeneous-cut Algorithm. $K=256, p=(8,8,8)$.

Figure 5.3 compares the average running time and WRMSE of all the activity weighted versions of the same divisive algorithms analyzed in Figure 5.2. Comparing Figure 5.3 to Figure 5.2 shows that the heterogeneous-cut algorithm is also successful when activity weighting is used. For the given test set, one could justifiably call a quantization technique high quality if the average WMSE is less than 5.0. Given this definition of high quality, the heterogeneous-cut algorithm is once again the clear winner with respect to time. Thus, of all the divisive algorithms studied by this dissertation, the heterogeneous-cut algorithm has obtained the best balance between time and quality.

Table 5.1 shows the number of splits made by the heterogeneous-cut algorithm and the percentage of splits made in each of the five categories. On average over 65% of the splits are in categories 2 or 3; however, Figure 5.2 and Figure 5.3 show that the average running times are closest to category 5 and the average RMSE and WRMSE are closest to category 4.

	# Of Splits	1: $\sigma^2 e_1 \perp_e$	2: $\sigma^2 E_{\downarrow t}$	3: $\sigma^2 \sigma^2 t$	4: $\sigma^2 \sigma^2 \mu$	5: $\sigma^2 8t$
Max	252	33.99%	50.29%	44.05%	22.22%	16.28%
Min	129	0.53%	13.49%	26.80%	1.55%	0.40%
Avg	174.76	8.84%	37.02%	34.78%	10.83%	8.53%
Max	253	43.38%	45.61%	43.25%	22.13%	16.28%
Min	129	0.00%	7.54%	19.12%	1.47%	0.40%
Avg	176.2	13.66%	34.38%	32.62%	10.96%	8.38%

Table 5.1: **Number of Splits Made by the Heterogeneous-Cut Algorithm.** $K=256$, $p=(8,8,8)$. The shaded data indicates activity weighting was used.

5.1.2 Subjective Observations

Color Plate 5.1 illustrates that indeed the proposed heterogeneous-cut algorithm produces high quality quantized images. Comparing Color Plate 5.1 to the images in Color Plate 3.4, Color Plate 3.5, Color Plate 3.6, and Color Plate 3.7 shows that the heterogeneous-cut algorithm is competitive with many of the high quality

quantization techniques. It clearly isn't the best performer, but nor is it the worst. Recall, however, it is amongst the fastest!



Color Plate 5.1: **Proposed Heterogeneous-Cut Quantization of *Solids* and *Lena*.** $K=256$, $p=(8,8,8)$.

5.2 Weighted MinMax

This dissertation conjectures that the MinMax algorithm produces quantized images with considerable contouring because it treats all colors equally. The RMSE (or

WRMSE) metric gives weight to each color proportional to the color's frequency (or activity weighting). Thus, this dissertation proposes modifying the MinMax algorithm in two simple ways:

- (i) the distance from the each color to its representative is weighted by the number of times that color occurs in the image, and
- (ii) the representative color of each cluster is set to its centroid at the end of each iteration (as in the LBG algorithm).

In this way, the colors that occur more frequently are given higher precedence. Pseudocode for the proposed weighted MinMax algorithm is provided in Figure 5.4.

Algorithm:	Weighted MinMax algorithm to select representatives
Input:	C The set of N unique colors in I
	K The numbers of colors to be selected.
	<i>Assumption:</i> $N > K$.
	H Histogram of I
Output:	R_K Selected representatives for I
	S_K K -split for C
	<i>// Step 1: Initialization</i>
1	$s_0 \leftarrow C = \{c_1, c_2, \dots, c_N\} \subseteq \text{RGB}$
2	$r_0 \leftarrow c_1$
	<i>// Step 2: Iterative Creation of New Clusters</i>
3	for $k \leftarrow 1$ upto $K - 1$
4	$\delta \leftarrow \max \{d_2(c_i, r_j) \cdot H(c_i) \mid c_i \in s_j \text{ and } 0 \leq j \leq k-1\}$
5	$c \leftarrow$ one of the colors whose distance to its representative is δ
6	move c to s_k
7	$r_k \leftarrow c$
	<i>// Step 3: Reassignment of Colors to New Cluster</i>
8	for each $j \leftarrow 0$ upto $k-1$
9	for each $c \in s_j$
10	if $d_2(c, r_k) \leq d_2(c, r_j)$
11	move c to s_k
12	for each $j \leftarrow 0$ upto k
13	$r_j \leftarrow$ centroid of s_j
14	return R_K and S_K

Figure 5.4: **Pseudocode for the Weighted MinMax Algorithm.** The shaded lines indicate changes from the original MinMax algorithm given in Figure 4.1.

Theorem 3.2 does not apply to the weighted MinMax algorithm because the weighted distance used in lines 4 and 10 of Figure 5.4 does not satisfy the triangle inequality and because the cluster representatives may change at the end of each iteration of the while loop.

5.2.1 Empirical Analysis

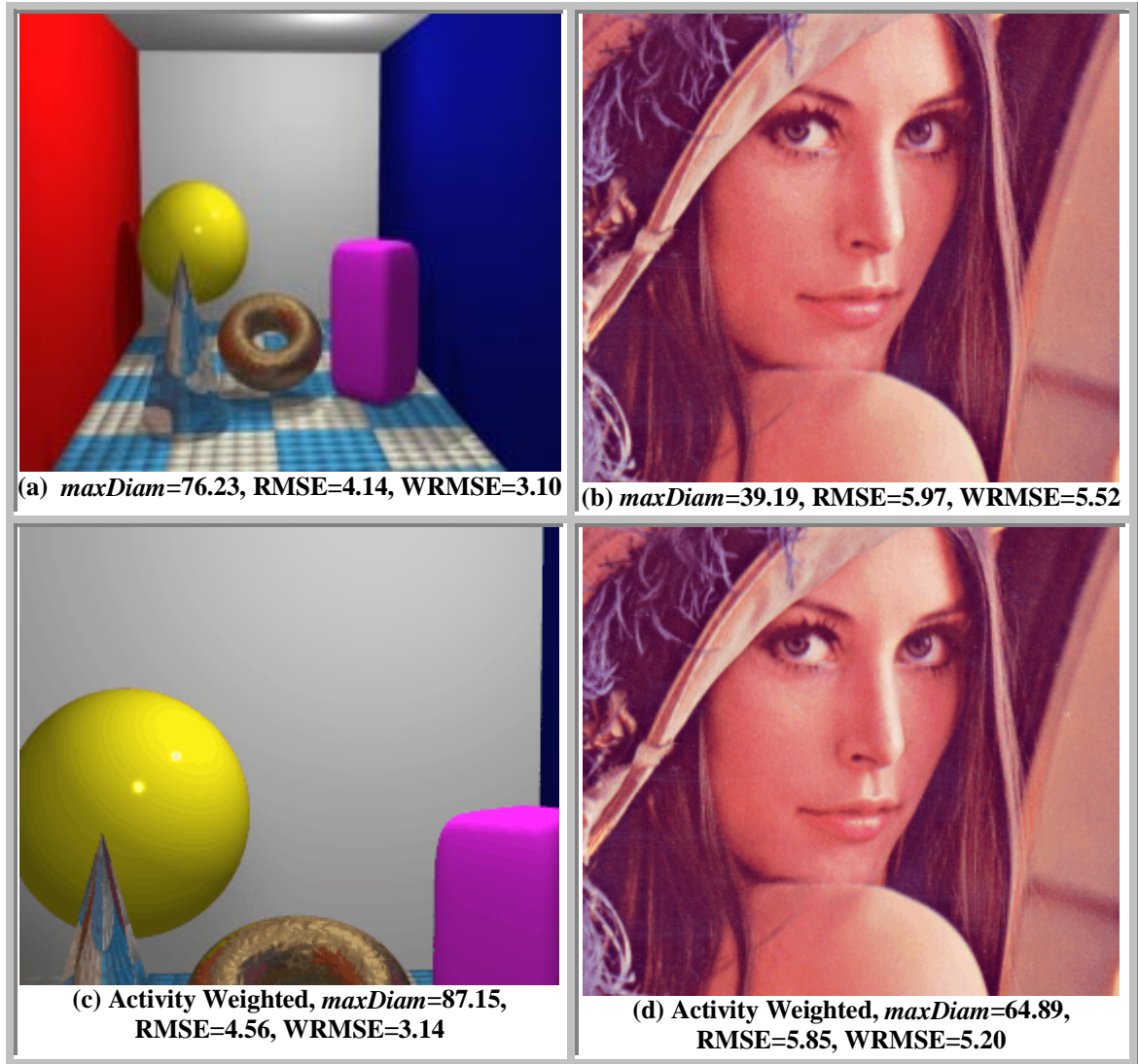
#	<i>Time (s)</i>	<i>maxDiam</i>	<i>RMSE</i>	<i>WRMSE</i>
Max	311.20	105.76	11.60	9.09
Min	9.83	37.20	1.84	1.58
Avg	86.43	62.36	5.77	4.73
Max	265.83	125.06	10.94	9.01
Min	9.98	37.20	2.00	1.60
Avg	91.95	76.29	6.01	4.68

Table 5.2: **Empirical Data for Proposed Weighted MinMax.** $K=256$, $p=(8,8,8)$. The shaded data indicates activity weighting was used.

Table 5.2 gives the *maxDiam*, RMSE, WRMSE and the maximum, minimum and average running time for each of the three versions of the proposed weighted MinMax algorithm. Comparing the average running time data in Table 4.1, Table 4.2, and Table 5.2 shows that the average running time of the weighted MinMax algorithm is about one and a half times that of MinMax and about half that of LBG. Using the definitions of “high quality” proposed in previous section, the proposed weighted MinMax algorithm passes muster on both accounts. The average RMSE for non-activity weighting is less than 6.0, and the average WRMSE is less than 5.0 for activity weighting. Note, however, Table 4.1 and Table 4.2 show that MinMax does not qualify as high quality, but LBG does. Thus, the proposed weighted MinMax algorithm illustrates how combining a “low quality” quantization technique (MinMax) with a high quality quantization technique (LBG) yields a high quality technique whose running time is in between the two techniques. However, since the average running time of proposed weighted MinMax is about four times greater than

the slowest hierarchically divisive algorithm analyzed in Section 3.3, it is not competitive with these algorithms from a time perspective.

5.2.2 Subjective Observations



Color Plate 5.2: **Proposed Weighted MinMax Quantization of *Solids* and *Lena*.**
 $K=256$, $p=(8,8,8)$.

Color Plate 5.2 illustrates the success of the proposed weighted MinMax algorithm in producing high quality quantized images. Comparing Color Plate 5.2 to the images in

Color Plate 4.2 and Color Plate 4.3 shows that the weighted MinMax-cut algorithm far superior to MinMax and is competitive with LBG.

5.3 Summary

This chapter proposed and analyzed two color quantization techniques that are hybrids of quantization techniques discussed in Chapter 3 and Chapter 4. Both of the proposed algorithms show how combining low quality, but fast color quantization techniques with high quality (but slower) techniques achieves new high quality methods with running times that are in between the slow and fast methods, but closer to the fast.

Chapter 6

Conclusions

This dissertation conducted a comprehensive survey and comparative analysis of PNN and thirty-eight hierarchically divisive color quantization techniques. The divisive algorithms were discussed within the framework of a general taxonomy for classifying divisive algorithms proposed by this dissertation. The analysis used a test set consisting of twenty-five diverse images, and is the most comprehensive comparative analysis conducted to date. PNN was shown to achieve quantized images with lower RMSE than the divisive algorithms; however, PNN requires an impractical amount of time to do so. Several of the divisive methods had not been previously studied.

This dissertation also described and analyzed three non-hierarchical algorithms. The analysis of the MinMax algorithm showed that the *maxDiam* metric is inappropriate for color image quantization. The LBG iterative refinement algorithm was shown to be a much more cost-effective way to achieve image quality close to PNN.

This dissertation proposed and analyzed two color quantization techniques that are hybrids of quantization techniques previously proposed in the literature. The first method adaptively combines five hierarchically divisive techniques, capitalizing on the strengths of each technique, but not suffering from their weaknesses. The second method combines MinMax and LBG. Both of the proposed algorithms show how combining low quality, but fast color quantization techniques with high quality (but slower) techniques achieves new high quality methods with running times that are in between the slow and fast methods, but closer to the fast. A generalized method for applying activity weighting to any histogram-based color quantization algorithm was also proposed. The generalized activity weighting was shown to be a fast and effective way to enhance the quality of quantized images.

REFERENCES

- AkOA97 AKARUN, LALE, DOGAN ÖZDEMİR, AND E. ALPAYDIN. "Fuzzy Error Diffusion of Color Images." *Proceedings of International Conference on Image Processing*, 3:46-9, Santa Barbara, California, (October 26-29, 1997).
- AkOY96 AKARUN, LALE, DOGAN ÖZDEMİR, AND ÖZER YALÇIN. "Modified Quantization Algorithm for Dithering of Colour Images." *Electronics Letters*, 32(13):1185-6, 1996.
- And73 ANDERBERG, MICHAEL R. *Cluster Analysis for Applications*. New York: Academic Press, 1973
- BaA91a BALASUBRAMANIAN, RAJA AND JAN P. ALLEBACH. "A New Approach to Palette Selection for Color Images." *Human Vision, Visual Processing, and Digital Display II*, pp. 58-69, San Jose, California, (February 27-March 1, 1991).
- BaA91b BALASUBRAMANIAN, RAJA AND JAN P. ALLEBACH. "A New Approach to Palette Selection for Color Images." *The Journal of Imaging Science and Technology*, 17(6):284-90, (December 1991).
- BaAB94 BALASUBRAMANIAN, RAJA, JAN P. ALLEBACH, AND CHARLES A. BOUMAN. "Color-Image Quantization with use of a Fast Binary Splitting Technique." *Journal of the Optical Society of America*, 11(11):2777-86, (November 1994).
- BaBA92 BALASUBRAMANIAN, RAJA, CHARLES A. BOUMAN, AND JAN P. ALLEBACH. "New Results in Color Image Quantization." *Image Processing Algorithms and Techniques III*, pp. 289-303 (February 10-13, 1992).

- Bra86 BRAUDAWAY, G. "Procedure for Optimum Choice of a Small Number of Colors from a Large Palette for Color Imaging." *IBM Technical Disclosure Bulletin*, 29(3):1329-34, (August 1986).
- BrB97 BRAQUELAIRE, JEAN-PIERRE AND LUC BRUN. "Comparison and Optimization of Methods of Color Image Quantization." *IEEE Transactions on Image Processing*, 6(7):1048-52, (July 1997).
- ChC93 CHANG, LONG-WEN AND HER-HSIUNG CHANG. "A Fast Algorithm for Color Vector Quantization." *Applications of Digital Processing XVI*, 2028:106-15, San Diego, California, (July 14-16, 1993).
- ChTM94 CHADDHA, NAVIN, WEE-CHIEW TAN, AND TERESA H.Y. MENG. "Color Quantization of Images Based on Human Visual Perception." *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, V:89-92, Adelaide, South Australia, (April 19-22, 1994).
- Cla95 CLARK, DEAN. "The Popularity Algorithm." *Dr. Dobb's Journal*, pp. 121-6, (July 1995).
- CoF97 CONNOLLY, CHRISTINE AND THOMAS FLIESS. "A Study of Efficiency and Accuracy in the Transformation from RGB to CIELAB Color Space." *IEEE Transactions on Image Processing*, 6(7):1046-7, (July 1997).
- Dal83 DALY, SCOTT. "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity," in *Digital Images and Human Vision*. Cambridge, Massachusetts: MIT Press, 1993.
- Dek94 DEKKER, ANTHONY H. "Kohonen Neural Networks for Optimal Colour Quantization." *Network: Computation in Neural Systems*, 5:351-67, 1994.
- Dix91 DIXIT, SUDHIR S. "Quantization of Color Images for Display/Printing on Limited Color Output Devices." *Computers and Graphics*, 15(4):561-7, 1991.

- Equ89 EQUITZ, WILLIAM H. "A New Vector Quantization Clustering Algorithm." *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(10):1568-75, (October 1989).
- FiO89 FIUME, E. AND M. OUELLETTE. "On Distributed, Probabilistic Algorithms for Computer Graphics." *Proceedings of Graphics Interface '89*, pp. 211-18, London, Ontario, (June 19-23, 1989).
- FIS76 FLOYD, ROBERT W. AND LOUIS STEINBERG. "An Adaptive Algorithm for Spatial Gray Scale." *Proceedings of the Society for Information Display*, 17(2):75-7, 1976.
- FoVFHP93 FOLEY, JAMES D., ANDRIES VAN DAM, JOHN F. HUGHES, AND RICHARD L. PHILLIPS. *Introduction to Computer Graphics*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1993.
- FrS97 FREISLEBEN, BERND AND ANDREAS SCHRADER. "An Evolutionary Approach to Color Image Quantization." *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 459-64, Indianapolis, Indiana, (April 13-16, 1997).
- GaJW82 GAREY, M.R., D.S. JOHNSON, AND HANS S. WITSENHAUSEN. "The Complexity of the Generalized Lloyd-Max Problem." *IEEE Transactions on Information Theory*, 28(2):255-6, (March 1982).
- GaMC94 GALLI, I., A. MECOCCHI, AND V. CAPPELLINI. "Improved Colour Image Vector Quantisation by Means of Self-Organizing Neural Networks." *Electronics Letters*, 30(4):333-4, (February 17 1994).
- GeAW90 GENTILE, RONALD S., JAN P. ALLEBACH, AND ERIC WALOWIT. "Quantization of Color Images Based on Uniform Color Spaces." *The Journal of Imaging Science and Technology*, 16(1):11-21, (February 1990).
- GeRS93 GEIST, ROBERT, ROBERT REYNOLDS, AND DARRELL SUGGS. "A Markovian Framework for Digital Halftoning." *ACM Transactions on Graphics*, 12(2):136-59, (April 1993).

- Gir83 GIROD, BERN. "What's Wrong with Mean-squared Error?," in *Digital Images and Human Vision*. Cambridge, Massachusetts: MIT Press, 1993.
- GoI91 GOLDBERG, NAFTALY. "Colour Image Quantization for High Resolution Graphics Display." *Image and Vision Computing*, pp. 303-12, 1991.
- Gon85 GONZALEZ, TEOFILO F. "Clustering to Minimize the Maximum Intercluster Distance." *Theoretical Computer Science*, 38(2-3):23-306, 1985.
- GoW92 GONZALEZ, RAFAEL C. AND RICHARD E. WOODS. *Digital Image Processing*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1992
- GrW91 GROGAN, TIMOTHY A. AND MEI WU. "Image Quality Measurements with Neural-Brightness Perception Model." *Human Vision, Visual Processing, and Digital Display II*, pp. 16-30, San Jose, California, (February 27-March 1, 1991).
- Hec80 HECKBERT, PAUL S. "Color Image Quantization for Frame Buffer Display." Bachelor's thesis, Massachusetts Institute of Technology, 1980.
- Hec82 HECKBERT, PAUL S. "Color Image Quantization for Frame Buffer Display." *Computer Graphics*, 16(3):297-304, (July 1982).
- HoD86 HOULE, G. AND E. DUBOIS. "Quantization of Color Images for Display on Graphics Terminals." *Proceedings IEEE Global Telecommunications Conference*, 2:1138-42, Houston, Texas, (December 1-4, 1986).
- Hul90 HULTGREN, BROR O. "Subjective Quality Factor Revisited." *Human Vision and Electronic Imaging: Models, Methods, and Applications*, pp. 12-22, Santa Barbara, California, (February 12-14, 1990).
- JaJN76 JARVIS, J.F., C.N. JUDICE, AND W.H. NINKE. "A Survey of Techniques for the Display of Continuous-Tone Pictures on Bilevel Displays." *Computer Graphics and Image Processing*, 5:13-40, 1976.

- JoX93 JOY, GREGORY AND ZHIGANG XIANG. "Center-cut for Color Image Quantization." *The Visual Computer*, 10:62-6, 1993.
- JoX96 JOY, GREGORY AND ZHIGANG XIANG. "Reducing False Contours in Quantized Color Images." *Computers and Graphics*, 20(2):231-42, (March/April 1996).
- KiLLH96a KIM, KYEONG-MAN, CHAE-SOO LEE, EUNG-JOO LEE, AND YEONG-HO HA. "Color Image Quantization using Weighted Distortion Measure of HVS Color Activity." *Proceedings of 3rd IEEE International Conference on Image Processing*, 3:1035-9, Lausanne, Switzerland, (September 16-19, 1996).
- KiLLH96b KIM, KYEONG-MAN, CHAE-SOO LEE, EUNG-JOO LEE, AND YEONG-HO HA. "Color Image Quantization and Dithering Method Based on Human Visual System Characteristics." *The Journal of Imaging Science and Technology*, 40(6):502-9, (November/December 1996).
- KoCL93 KOK, C.W., S.C. CHAN, AND S.H. LEUNG. "Color Quantization by Fuzzy Quantizer." *Nonlinear Image Processing IV*, pp. 235-42 (February 1-3, 1993).
- Knu86 KNUTH, DONALD E. "Digital Halftones by Dot Diffusion." *ACM Transactions on Graphics*, 6(4):245-73, 1987.
- LeTT96 LEMSTRÖM, KJELL, JORMA TARHIO, AND TAPIO TAKALA. "Color Dithering with n-best Algorithm." *Proceedings of WSCG 96: Fourth International Conference in Central Europe on Computer Graphics & Visualization*, 1:162-9, Plzen, Czech Republic, (February 12-16, 1996).
- LiBG80 LINDE, YOSEPH, ANDRÉS BUZO, AND ROBERT M. GRAY. "An Algorithm for Vector Quantizer Design." *IEEE Transactions on Communications*, 28(1):84-95, 1980.
- LiC94 LIU, TSANN-SHYONG AND LONG-WEN CHANG. "Greedy Tree Growing for Color Image Quantization." *Proceedings of IEEE International*

Conference on Acoustics, Speech and Signal Processing, V:97-100, Adelaide, South Australia, (April 19-22, 1994).

- LiC95 LIU, TSANN-SHYONG AND LONG-WEN CHANG. "Fast Color Image Quantization with Error Diffusion and Morphological Operations." *Signal Processing :The Official Publication of the European Association for Signal Processing (EURASIP)*, 43(3):293-303, (May 1995).
- Lim79 LIMB, JOHN O. "Distortion Criteria of the Human Viewer." *IEEE Transactions on Systems, Man and Cybernetics*, 9(12):778-93, (December 1979).
- Lu91 LU, HONG-QIAN. "Quantitative Evaluation of Image Enhancement Algorithms." *Human Vision, Visual Processing, and Digital Display II*, pp. 223-34, San Jose, California, (February 27-March 1, 1991).
- MuV94 MURRAY, JAMES D. AND WILLIAM VANRYP. *Encyclopedia of Graphics File Formats*. Sebastopol, CA: O'Reilly & Associates, Inc., 1994
- OrB91 ORCHARD, MICHAEL T. AND CHARLES A. BOUMAN. "Color Quantization of Images." *IEEE Transactions on Acoustics, Speech and Signal Processing*, 39(12):2677-90, (December 1991).
- Ots79 OTSU, NOBUYUKI. "A Threshold Selection Method for Grey Level Histograms." *IEEE Transactions on Systems, Man and Cybernetics*, 9:62-6, (January 1979).
- PaLKLH96 PARK, YANG-WOO, EUNG-JOO LEE, KYEONG-MAN KIM, CHE-SOO LEE, AND YEONG-HO HA. "Color Enhancement by New Quantization of Fast Clustering Method." *Proceedings of Imaging Science and Technology: Evolution and Promise*, pp. 275-81, Chiba, Japan, (September 11-14, 1996).
- PeL98 PEI, SOO-CHANG AND YOU-SHEN LO. "Color Image Compression and Limited Display using Self-organization Kohonen Map." *IEEE Transactions on Circuits and Systems for Video Technology*, 8(2):191-205, (April 1998).

- Rei97 REITAN, PAULA J. "3D Visualization of Truecolor Image Histograms." *Sixth International Conference on Computational Graphics and Visualization Techniques*, pp. 320-9, Vilamoura, Algarve, Portugal, (December 15-18, 1997).
- Rei98 REITAN, PAULA J. "3D Visualization of Color Image Histograms." *Computer Network and ISDN Systems*, 30(20-21):2025-35, (November 1998).
- ReSMG97a RENDÓN, ENRIQUE, LUIS SALGADO, JOSÉ MENENDEZ, AND NARCISO GARCÍA. "Adaptive Palette Determination for Color Images based on Kohonen Networks." *Proceedings of International Conference on Image Processing*, 1:830-3, Santa Barbara, California, (October 26-29, 1997).
- ReSMG97b RENDÓN, ENRIQUE, LUIS SALGADO, JOSÉ MENENDEZ, AND NARCISO GARCÍA. "Adaptive Color Quantization through Self-Organization Neural Networks." *Proceedings of 4th International Workshop on Systems, Signals and Image Processing*, pp. 195-8,248, Poznan, Poland, (May 28-30, 1997).
- RoG95 ROYTMAN, EVGENY AND CRAIG GOTSMAN. "Dynamic Color Quantization of Video Sequences." *IEEE Transactions on Visualization and Computer Graphics*, 1(3):274-86, (September 1995).
- Sak77 SAKRISON, DAVID J. "On the Role of the Observer and a Distortion Measure in Image Transmission." *IEEE Transactions on Communications*, 25(11):1251-67, (November 1977).
- Say96 SAYOOD, KHALID. *Introduction to Data Compression*. San Francisco, California: Morgan Kaufmann Publishers, Incorporated, 1996
- ScD97 SCHEUNDERS, P. AND S. DE BACKER. "Joint Quantization and Error Diffusion of Color Images using Competitive Learning." *Proceedings of International Conference on Image Processing*, 1:811-14, Santa Barbara, California, (October 26-29, 1997).

- Sch96a SCHEUNDERS, P. "A Genetic Approach Towards Optimal Color Image Quantization." *Proceedings of 3rd IEEE International Conference on Image Processing*, 3:1031-4, Lausanne, Switzerland, (September 16-19, 1996).
- Sch96b SCHEUNDERS, P. "A Genetic Lloyd-Max Image Quantization Algorithm." *Pattern Recognition Letters*, 17(5):547-56, (May 1 1996).
- Sch97 SCHEUNDERS, P. "A Genetic c-means Clustering Algorithm Applied to Color Image Quantization." *Pattern Recognition*, 30(6):859-66, (June 1997).
- Shu95 SHUFELT, JEFFEREY A. "Color Image Quantization Enhancement Techniques." Carnegie Mellon University: Computer Science Department, TR CMU-CS95-100, (September 1995).
- Shu97 SHUFELT, JEFFEREY A. "Texture Analysis for Enhanced Color Image Quantization." *Graphical Models and Image Processing*, 59(3):149-63, (May 1997).
- Spr91 SPROULL, ROBERT F. "Refinements to Nearest-Neighbor Searching in k-Dimensional Trees." *Algorithmica*, 6:579-89, 1991.
- StFB92 STOKES, MIKE, MARK D. FAIRCHILD, AND ROY S. BERNIS. "Precision Requirements for Digital Color Reproduction." *ACM Transactions on Graphics*, 11(4):406-22, (October 1992).
- TaAE98 TASDIZEN, TOLGA, LALE AKARUN, AND CEM ERSOY. "Color Quantization with Genetic Algorithms." *Signal Processing: Image Communication*, 12:49-57, 1998.
- TrCC97 TRÉMEAU, ALAIN, C. CHARRIER, AND H. CHERIFI. "A Vector Quantization Algorithm based on the Nearest Neighbor of the Furthest Color." *Proceedings of International Conference on Image Processing*, 3:682-5, Santa Barbara, California, (October 26-29, 1997).

- Uli88 ULICHNEY, ROBERT A. "Dithering with Blue Noise." *Proceedings of the IEEE*, 76:56-79, (January 1988).
- VeB95 VEREVKA, OLEG AND JOHN W. BUCHANAN. "Local K-means Algorithm for Colour Image Quantization." *Proceedings of Graphics Interface '95*, pp. 128-35, Québec City, Québec, Canada, (May 17-19, 1995).
- VeGS97 VELHO, LUIZ, JONAS GOMES, AND MARCOS VINICIUS RAYOL SOBREIRO. "Color Image Quantization by Pairwise Clustering." *Proceedings X Brazilian Symposium on Computer Graphics and Image Processing*, pp. 203-10, Campos do Jordao, Brazil, (October 14-17, 1997).
- WaPW90 WAN, S.J., P. PRUSINKIEWICZ, AND S.K.M. WONG. "Variance-Based Color Image Quantization for Frame Buffer Display." *COLOR Research and Applications*, 15(1):52-8, (February 1990).
- War63 WARD, JOE H. "Hierarchical Grouping to Optimize an Objective Function." *Journal of the American Statistical Association*, 58(301):236-44, (March 1963).
- WaWP88 WAN, S.J., S.K.M. WONG, AND P. PRUSINKIEWICZ. "An Algorithm for Multidimensional Data Clustering." *ACM Transactions on Mathematical Software*, 14(2):153-62, (June 1988).
- Wu91a WU, XIAOLIN. "Statistical Colour Quantization for Minimum Distortion." *Computer Graphics and Mathematics*, pp. 189-202, Genoa, Italy, (October, 1991).
- Wu91b WU, XIAOLIN. "Efficient Statistical Computations for Optimal Color Quantization," in *Graphics Gems II*. New York: Academic Press, 1991.
- Wu92 WU, XIAOLIN. "Color Quantization by Dynamic Programming and Principal Analysis." *ACM Transactions on Graphics*, 11(4):348-72, (October 1992).

- WuW85 WU, XIAOLIN AND IAN H. WITTEN. "A Fast k-means Type Clustering Algorithm." University of Calgary, Canada: Department of Computer Science, TR 85/197/10, (May 1985).
- WuZ91 WU, XIAOLIN AND KAIZHONG ZHANG. "A Better Tree-Structured Vector Quantizer." *IEEE Data Compression Conference*, pp. 392-401 1991.
- WyS82 WYSZECKI, G. AND W. STILES. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Second Edition. New York: John Wiley & Sons, 1982
- Xia97 XIANG, ZHIGANG. "Color Image Quantization by Minimizing the Maximum Intercluster Distance." *ACM Transactions on Graphics*, 16(3):260-76, (July 1997).
- XiJ94a XIANG, ZHIGANG AND GREGORY JOY. "Feedback-based Quantization of Color Images." *Image and Video Processing II*, pp. 34-42, San Jose, California, (February 7-9, 1994).
- XiJ94b XIANG, ZHIGANG AND GREGORY JOY. "Color Image Quantization by Agglomerative Clustering." *IEEE Computer Graphics and Applications*, 14:44-8, (May 1994).
- XiJ96 XIANG, ZHIGANG AND GREGORY JOY. "Reducing False Contours in Quantized Color Images." *Computers and Graphics*, 20(2):231-42, 1996.
- YaL94 YANG, CHING-YUNG AND JA-CHEN LIN. "Use of Radius Weighted Mean to Cluster Two-class Data." *Electronics Letters*, 30(10):757-9, (May 12 1994).
- YaL95 YANG, CHING-YUNG AND JA-CHEN LIN. "Color Quantization by RWM-Cut." *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 2:669-72, Montreal, Québec, Canada, (August 14-16, 1995).
- YaL96 YANG, CHING-YUNG AND JA-CHEN LIN. "RWM-Cut for Color Image Quantization." *Computers and Graphics*, 20(4):577-88, 1996.